Document Clustering in Large German Corpora Using Natural Language Processing

Thesis

presented to the Faculty of Arts

of the University of Zurich

for the degree of Doctor of Philosophy

by

Richard Forster

of Neunkirch (SH)

Accepted on the recommendation of

Professor Dr. Michael Hess and

Professor Abraham Bernstein Ph.D.

2006

Abstract

Ever since the advent of computer systems and, in particular, the Internet, the amount of information *theoretically* at our disposal has been increasing exponentially. This phenomenal growth in data is not only a blessing: the more information is available, the more difficult it becomes to find one's way to the particular piece of information of interest. As a consequence, investigations into old and new techniques for dealing with the extraordinary flood of data remain topical for information science.

Cluster analysis has a rich and independent history of its own. Relatively recently it has acquired two new application areas in the fields of Information Retrieval and Data Mining. Clustering is used here to divide large unstructured document corpora into groups of more or less closely related documents. The clusters can then be used as a well-arranged interface to a potentially huge and overwhelming number of documents, allowing a prospective user to home in quickly on his specific requirements.

Document clustering consists of two fundamental stages: *document representation* (the transformation of documents as linear strings of words into suitable data structures) and *clustering* (the algorithmic grouping of these representations). Despite a number of detailed textbooks on cluster analysis in general, no such work seems to have been carried out on the specific needs of document clustering. Special attention is therefore paid to a comprehensive introduction. The first part of the thesis is dedicated to systematic surveys of existing clustering algorithms (with emphasis on those used for documents) and document representation techniques as encountered in practice. Particular care has been taken with the presentation of a uniform notation since the cluster analysis literature is notoriously rich in notations and multiple names for, often, one and the same concept. In addition, a scheme is presented for classifying different clustering applications in accordance with their time-criticality.

The second part of the thesis is devoted to an evaluation of Natural Language Processing (NLP) as a means of improving the document representations. More generally, the goal has been to help answer the old key question of whether or not the extra effort usually required for sophisticated NLP applications is rewarded by sufficient extra benefits; document clustering provides one typical battle-ground.

To this end, five large German data sets have been compiled and described. Each consists of several thousand documents and each was extracted from a specific source (two news outlets, two book services and one encyclopaedic data base). In each data set the documents carried separate content labels derived from meta-information provided by the original source. Serving as "objective truths", these labels were used to evaluate the performance of different clustering algorithms and document representations.

NLP techniques ranging from the very simple to complex syntactic and semantic models were then tested and evaluated on these five data sets. The techniques were divided into two groups: those aiming at a reduction of the representation complexity (with the twofold goal of achieving qualitative improvements and quantitative savings) and those aiming to enhance document representation with extra features (the goal now only being to improve the results through a more refined representation). Separately, there followed an evaluation of how well the various techniques worked together.

The thesis ends with an interpretation of the results, a discussion of the virtues shown by NLP methods in this particular domain and an overview of future research areas. It emerges that the success of many NLP representation techniques depends on the data under consideration, for which a hypothetical explanation is offered. All in all, evidence is found both *pro* and *contra* Natural Language Processing. For the majority of individual cases, distinct potential for improvement through NLP can be shown.

Zusammenfassung

Seitdem die Computer und das Internet in unseren Alltag getreten sind, hat die Informationsmenge, zu der wir *theoretisch* Zugang haben, exponentiell zugenommen. Dieses Wachstum ist nicht nur ein Segen; denn je mehr Informationen uns zur Verfügung stehen, desto schwieriger wird es, genau die gewünschte Einzelinformation zu finden. Die Erforschung von alten und neuen Strategien zur Bewältigung dieser Datenflut steht deshalb noch immer im Zentrum der Informationswissenschaften.

Die sogenannte *Clusteranalyse* blickt auf eine abwechslungsreiche Geschichte zurück, doch wurde sie erst vor relativ kurzem für das "Information Retrieval" und "Data Mining" entdeckt. Sie wird hier dazu gebraucht, grosse unstrukturierte Textmengen in Gruppen oder Haufen ("Clusters") von mehr oder weniger stark miteinander verwandten Dokumenten zu unterteilen. Daraus lassen sich einfache und übersichtliche Schnittstellen zu potentiell riesigen Korpora gewinnen, die es dem Anwender erlauben, schnell zu den für ihn relevanten Texten vorzustossen.

Text-Clustering besteht aus zwei grundlegenden Schritten: der *Text-Repräsentation* (Umwandlung von Texten als Zeichenketten in geeignete Datenstrukturen) und dem *Clustering* (Analyse dieser Repräsentationen und Ordnung in Gruppen). Trotz umfangreicher Literatur zur Clusteranalyse fehlt ein eigenständiges Lehrbuch zum Text-Clustering, weshalb in der vorliegenden Arbeit besonderer Wert auf eine umfassende Einleitung gelegt wurde. Der erste Teil der Dissertation besteht aus einer systematischen Übersicht über die Vielfalt der Cluster-Algorithmen (mit Schwerpunkt auf den für das Text-Clustering interessanten Methoden) und über die geläufigen Text-Repräsentationsmethoden. Angesichts der zahlreichen Synonyme und vielfältigen Notationen in der Clusteranalyse-Literatur wird dabei speziell auf eine einheitliche Notation geachtet. Anschliessend wird ein Schema zur Klassifikation von Text-Clustering-Anwendungen eingeführt, das sich an den zeitkritischen Komponenten orientiert.

Der zweite Teil der Dissertation untersucht die Verwendung Natürlichsprachlicher Datenverarbeitung (Natural Language Processing – NLP) bei der Text-Repräsentation. Auf einer höheren Ebene geht es dabei um die Frage, ob der oft beträchtliche Zeitaufwand für NLP durch qualitative Gewinne gerechtfertigt werden kann, mit Text-Clustering als typischem Fallbeispiel.

Zu diesem Zweck werden fünf grosse Textsammlungen in deutscher Sprache zusammengestellt und beschrieben. Jede besteht aus mehreren Tausend Dokumenten und entstammt einer spezifischen Quelle (zwei Nachrichten-Korpora, zwei Büchervertriebe und eine Enzyklopädie). In jeder Sammlung werden die Dokumente anhand vorhandener Angaben bestimmten Klassen zugeordnet. Diese Klassen gelten fortan als "objektive Wahrheit" und dienen der Evaluation verschiedener Clustering-Ergebnisse.

NLP-Techniken aller Art werden über den fünf Sammlungen zur Anwendung gebracht und evaluiert. Zwei grosse Gruppen von Methoden werden hierbei unterschieden: auf der einen Seite die "Reduktionstechniken", mit der doppelten Zielsetzung, die Datenkomplexität zu vermindern und die Clustering-Qualität zu steigern, und auf der anderen Seite die "Erweiterungstechniken", die neue zusätzliche Textmerkmale generieren und ausschliesslich der Clustering-Qualität verpflichtet sind. Mit separaten Experimenten wird anschliessend untersucht, wie gut die verschiedenen Techniken miteinander kombiniert werden können.

Die Dissertation schliesst mit einer Interpretation der Ergebnisse, einer Diskussion des Wertes Natürlichsprachlicher Datenverarbeitung (NLP) im gegebenen Kontext und einem Ausblick auf offene Forschungsfragen. Es zeigt sich, dass der Erfolg vieler NLP-Methoden vom jeweiligen Korpus abhängt, wofür hypothetische Erklärungen formuliert werden. Insgesamt fördert die Untersuchung Ergebnisse zutage, die sowohl *für* wie *wider* den Einsatz von NLP sprechen. Für die Mehrheit der untersuchten Fälle kann jedoch ein deutliches Verbesserungspotential durch Natürlichsprachliche Datenverarbeitungsmethoden gezeigt werden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to Document Clustering

There is a tsunami of data that is crashing
onto the beaches of the civilised world.
This is a tidal wave of unrelated, growing data
formed in bits and bytes, coming in an unorganised,
uncontrolled, incoherent cacophony of foam.
It is filled with flotsam and jetsam.
It is filled with the sticks and bones and shells
of inanimate and animate life.
None of it is easily related,
none of it comes with any organisational methodology.
(...) The tsunami is a wall of data—
data produced at greater and greater speed,
greater and greater amounts to store in memory,
amounts that double, it seems, with each sunset.
On tape, on disks, on paper,
sent by streams of light.
Faster and faster,
more and more and more.

Richard Saul Wurman (*Information Architects*, 1996)

Following the tragic natural catastrophe in December 2004 in East Asia the occasionally encountered term "information tsunami" has assumed a new ring, and for reasons of piety it had probably better be dropped altogether from information scientists' vocabulary. For the moment, however, it is worthwhile to dwell a little longer on the parallels between the lethal oceanic phenomenon and its comparatively harmless counter-part in cyberspace. In both cases we are dealing with phenomena of absolutely staggering dimensions having dramatic implications for our lives. In both cases the consequences exceed everything known before, beating all human anticipations by far. In both cases there is no means of preventing or containing the phenomenon—all we can do is look out for strategies and techniques to cope with it as well as possible and to keep the damage to a minimum.

With the advent of the Internet the mass of data daily pouring onto us or at least standing at our free disposal has started to grow exponentially. And it keeps growing. A recent study estimated the size of the "indexable" Web at over 11.5 billion pages (Gulli and Signorini, 2005) and there is no end in sight. In fact, as entire libraries are undergoing a process of digitisation, we may have seen just the beginning.

This unimaginably huge flow of data is having a far-reaching impact on our societies, economies and daily lives. The immense quantity of available data offers us not only an unprecedented wealth of riches and possibilities, it also confronts us with a multitude of challenges and problems not encountered before. Because coping with the data is not trivial.

The dangers accompanying the surge of information are not to be underestimated and are well illustrated by buzzwords such as "Infoglut" (Allen, 1992), "Information Fatigue Syndrome" (Lewis, 1996), "TechnoStress" (Weil and Rosen, 1997), "Data Smog" (Shenk, 1997), "Data Asphyxiation" (Winkle, 1998) and "Information Pollution" (Nielsen, 2003). Whole new fields of psychology have been opened just to deal with the information overload and with our difficulties and perplexity resulting therefrom.

At the other end, an explosion of new research into fields such as data mining, information storage, information retrieval, knowledge extraction and knowledge management has set in. It is directed at developing the new tools that are desperately needed to cope with one of the big challenges of the 21st century: the information challenge.

The present thesis cannot but deal with a very tiny aspect of the global endeavour to fight the information flood. First, we restrict ourselves to *textual data*. Second, of the dozens (if not hundreds) of strategies and techniques that are being developed to cope with the information load, we pick out two: *document clustering* and *natural language processing* (NLP). We subject them to an in-depth study and try to assess the benefits of their combined application. The results should offer guidance to future investigations and other combinations of NLP with information retrieval techniques.

The rest of the present chapter is divided as follows: the first two sections motivate and introduce document clustering and natural language processing, giving as well an overview of the structure of the thesis (Sections 1.1 and 1.2). The next section presents a few central concepts in some more detail (Section 1.3). Two further sections deal with the purpose and applications of document clustering in theory and praxis (Sections 1.4 and 1.5), while the final section introduces the formal notation used in the following chapters (Section 1.6).

1.1 Motivation

The present thesis focusses on two aspects of automated information processing and their combined application:

- **Document Clustering** as an approach to bring order into large *sets of unordered documents*,
- **Natural Language Processing** as an approach to extract good descriptions of *individual documents*.

In particular, it is the goal of this thesis to establish *which natural language processing techniques are useful in a pre-processing step and to which extent they can improve clustering results.* The work will be further characterised by the choice of five independent, diverse and relatively large sets of German documents that have been specifically collected for the present study.

Cluster analysis is an old and well-established procedure for bringing a general order into large sets of all kind of data, including texts. In information retrieval, it has long remained in the background because of the predominant pursuit of powerful *search techniques* (i. e. ways to find particular documents or documents about a particular topic). But with the development of adequate solutions to the straightforward search problem (as demonstrated by Google[1] and other leading search companies), more complex information handling strategies, including document clustering, have received a new boost.

Natural language processing has its own long history with a wide number of applications such as machine translation, speech generation, question-answering systems, text summarisation and many more. Applied to the document clustering problem, NLP is used to replace the traditional view of a document as a random conglomerate of letters and digits (words) by an analytical linguistic view of these symbols. We want to show that an accordingly refined representation of documents in terms of linguistic concepts leads to superior clustering results, and in particular we want to find out which NLP techniques are the most suitable for the task.

The application of NLP techniques in information retrieval (IR) has frequently been discussed in the past and it remains a controversial issue.[2] In the end, it all boils down to a classic trade-off problem: do the extra benefits warrant the extra effort that is required by NLP? Notable experts such as Smeaton (1997) and Sparck Jones (1999) have taken a rather critical view of the general use of NLP in text retrieval systems. By narrowing the discussion to one specific aspect of information retrieval and subjecting our data to a thorough examination by an arsenal of different NLP techniques, we hope to be able to reach a more reliable verdict, even if only for one particular area of IR. A further aim is to characterise the circumstances under which the use of NLP is recommendable.

1.2 Thesis Structure

The structure of this thesis is governed by the two main components of a document clustering system (see Figure 1.1):

- The **document representation component**, which takes as input the documents (raw texts, Web pages, etc.), then extracts pertinent features and transforms them into a data structure that is suitable for clustering. Normally, a vector representation is chosen for the individual documents, resulting in an $n \times m$ document-feature matrix for the whole set. Element (i, j) of the matrix then indicates the strength/presence of feature j in document i.

- The **cluster analysis component**, which takes as input the data structure generated in the previous step and groups similar rows (documents) together. Typically, n data points are thus partitioned into k clusters. Cluster analysis is independent of the original domain (documents) and has applications in the most diverse areas of science. Its basic ingredients are a measure of (dis)similarity between data points and an algorithm for grouping them.

Depending on the application, the clusters thus gained can be further processed, e. g. by a *cluster visualisation component*. Some of these post-processing approaches will be touched upon in the text, but they are mostly outside the scope of this study. The goal is to integrate NLP techniques into the document representation component against the background of document cluster analysis.

The thesis is divided into eight chapters as follows:

[1] `www.google.com`

[2] See, for instance, Strzalkowski (1999), Feldman (1999), Zhou and Zhang (2003).

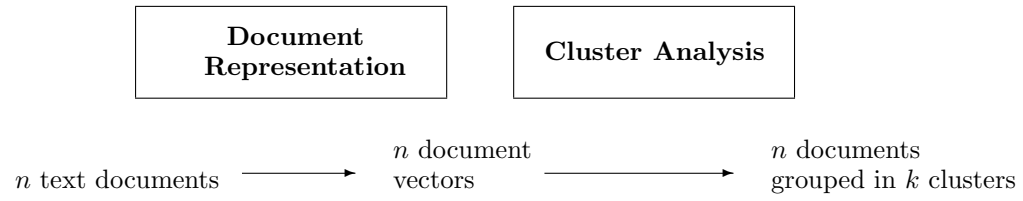


Figure 1.1: **The two main components of a document clustering system.** The *document representation component* maps text documents onto a suitable internal structure (usually a vector); the *cluster analysis component* determines a suitable grouping based on the input matrix built from the document vectors.

Chapter 1 (Introduction): The rest of the present chapter describes various applications and purposes of document clustering. The reader is introduced to the different situations and contexts in which clustering is used to organise text. A brief, annotated list of on-line resources is also given. Finally, the main concepts are introduced formally as a preparation for the following two theoretical chapters.

Chapter 2 (Cluster Analysis): First, we get acquainted with the main principles, data structures and algorithms that are used in the Clustering Component. We learn to know typical similarity measures, the main clustering algorithms (iterative, hierarchical and some others) and evaluation and validation techniques. Even though the discussion is kept at a fairly general level, special attention is paid to those clustering issues that are relevant to the domain of textual cluster analysis.

Chapter 3 (Document Representation): We then turn to the Document Representation component. We learn more about the mapping of documents onto data structures and about the processes of feature weighting, selection, standardisation and extraction. We introduce statistic and linguistic methods and conclude with a brief section on time constraints and cluster presentation.

Chapter 4 (Experimental Setup): After disseminating the theoretical background we turn to the experimental setup. We describe the five corpora used in the further experiments, the evaluation procedure for different representation methods and the clustering software.

Chapter 5 (Reduced Representations Using NLP): The first part of the experiments is devoted to representation *reduction* techniques. We examine different methods (including stopword removal, part-of-speech selection and several others) to filter and reduce the document representations. The aim is to reduce clustering complexity and to increase clustering quality.

Chapter 6 (Enhanced Representations Using NLP): We then proceed to representation *enhancement* techniques. We examine how morphological, syntactic and semantic information can be used to achieve richer and better document representations. The focus here is on improving clustering quality without unnecessarily increasing clustering complexity.

Chapter 7 (Combining Representation Techniques): After having analysed all techniques separately, we examine how well they combine with each other. We first look

at reduction and enhancement techniques among themselves, and then at selected combinations from both groups.

Chapter 8 (Summary and Outlook): The last chapter concludes with a summary of our findings and a brief analysis of the implications for Natural Language Processing. An outlook points to further research tasks that appear most rewarding in the light of the present study.

The thesis is rounded off by an appendix consisting of a glossary, a proof section, the stoplists and a detailed numeric account of the experimental results.

1.3 Principal Concepts

Let us characterise some of the principal concepts of the further discussion in a few more words:

Natural Language Processing (NLP). Natural Language Processing encompasses all those theories, hypotheses and practical applications which are directed at automatically processing text *based on knowledge of language, linguistics and human interaction.* The history of NLP dates back to the very beginning of the computer age and has been a central aspect of linguistic research and artificial intelligence ever since.

Jurafsky and Martin (2000, 4), after a brief overview of NLP history, distinguish six central areas of language processing:

- *phonetics and phonology:* the study of sounds,
- *morphology:* the study of meaningful components of words,
- *syntax:* the study of structural relationships between words,
- *semantics:* the study of meaning,
- *pragmatics:* the study of language used as a means to achieve certain goals,
- *discourse:* the study of complex linguistic interactions.

The present work features NLP techniques from the fields of morphology, syntax and semantics.

Information Retrieval (IR). Information retrieval is the science aiming to provide end-users with efficient access methods to large collections of data, usually in the form of documents. A concise definition is lacking, but the following quotations capture the essence:

- "Information retrieval is concerned with the representation, storage, organization, and accessing of information items. ... In actuality, many of the items found in ordinary retrieval systems are characterized by an emphasis on narrative information. Such narrative information must be analyzed to determine the information content and to assess the role each item may play in satisfying the information needs of the system users."—Salton and McGill (1983, 1–2).
- "Information retrieval is the term conventionally ... applied to the type of activity discussed in this volume. An information retrieval system does not inform (i. e. change the knowledge of) the user on the subject of his inquiry. It merely informs on the existence (or non-existence) and whereabouts of documents relating to his request."—Lancaster (1968, quoted in van Rijsbergen, 1979, 1).

1. User has information need.
2. User transforms information need into *query*. A query normally consists of one or more *query terms* explicitly or implicitly combined by Boolean operators.
3. IR system processes query. It matches the query to the documents in the collection and returns the best matches.
4. User is given a list of document references, often sparsely annotated and sorted by relevance (as perceived by the system).
5. User pursues one or more documents and/or modifies the query in the hope of obtaining better results (→ returns to step three).
6. User's information need is satisfied after reading the documents, and user is happy. As new information needs arise, he returns to step one.

Figure 1.2: **The standard scenario of information retrieval.**

- "Information retrieval [is] an area of science and technology that deals with cataloging, categorization, classification, and search of large amounts of information, particularly in textual form. An outcome of an information retrieval process is usually a set of documents containing information on a given topic, and may consist of newspaper-like articles, memos, reports of any kind, entire books."—Strzalkowski (1999, xiii).
- "Information retrieval is the task of finding relevant documents from a text corpus or collection in response to a user's information need."—Smeaton (1997, 15).

The standard IR scenario is described in Figure 1.2. In order to work, it requires powerful algorithms and storage methods. Often it is further refined by extra steps, feedback-loops, etc.

IR has been a primary concern of computer scientists since the 1960s, and it continues to play an extraordinary role which is well demonstrated by the unique importance of search engines for our own daily "Web behaviour" as well as for the marketing specialists and the emerging "search engine optimisation" industry.

Text/Document Retrieval. As we have just seen, information retrieval is concerned with bringing documents to users' attention. Whether the documents contain the information actually sought cannot be determined, though, and it is up to the user to extract and assess the information from the documents. Naturally, it would be even more desirable for systems to handle actual pieces of information and extract and present them rather than dealing only in coarse documents. In order to distinguish between such as yet mostly experimental systems and those in use at present, it has been suggested that today's ordinary search systems had better be described as *text* or *document* retrieval systems (van Rijsbergen, 1979, 1), with the therm *information retrieval* reserved for system that will be really capable of handling information. In the present work, however, the terms information retrieval, document retrieval and text retrieval will be used as synonyms.

Document Clustering. A document clustering system analyses a large set of documents and then groups similar documents together. As a result it produces a number of more or less well-defined *clusters* (groups) of documents. The process autonomously creates distinct

groups based on inter-document similarities. It thus imposes an order onto the previously unordered documents. Ideally, this order corresponds to a structure already inherent but not trivially discernible in the data.

The clusters may be entirely independent of each other. Alternatively, there may also be an order between the clusters. Some clustering systems produce an hierarchical order while others arrange clusters in a two- or three-dimensional space.

Clustering is an *unsupervised learning task*, as no *a priori* information is required.

Document Categorisation/Classification. Categorisation and classification (henceforth used synonymously) are strongly related to clustering. In both cases a potentially very large number of objects is to be grouped into a scheme of flat or hierarchical "classes". Yet, there are important differences between the classification/categorisation task and the clustering task (Willett, 1988):[3]

Classification is the process of assigning one or several objects to a *predefined scheme*. In other words, the individual classes are all given *a priori* and the task is to determine the best-fitting class for each new object. The classes have to be determined by the user, either intensionally (e. g. by a Boolean expression on a number of features that completely describe the class) or extensionally (by giving a sample of members for each class). Training an algorithm to recognise an extensional classification scheme is a *supervised learning task*; the algorithm is tuned on a pre-labelled *training set*.

Our focus is strictly on *clustering* (i. e. with no pre-defined classes), but the close relationship makes it unavoidable from time to time to compare our findings with those of the classification literature. Even though classification uses similar or identical data structures, the algorithms are often very different from those used for clustering.

Text Data Mining. Text data mining is a sub-discipline of data mining. It is the task of automatically extracting and processing information from textual data. Text data mining is a very wide but relatively unexplored and far less structured field than text retrieval (Hearst, 1999a; Williams, 2000a,b). In the long run it will no doubt play a key role in the global quest to accommodate the information avalanche. As with IR techniques, the question also arises how to make optimal use of NLP techniques for text mining.

1.4 Purpose and Applications of Document Clustering

Document clustering has been used for such diverse text data mining tasks as automatic generation of Frequently Asked Questions (FAQs) (Wen *et al.*, 2001; Wen and Zhang, 2003) and automatic functional annotation of gene products (Renner and Aszódi, 2000), as well as for many other tasks. By far the most typical applications of document clustering, however, have so far been found in information retrieval and they will be discussed in the rest of this section.

1.4.1 Supporting Retrieval of Documents

Historically speaking, document clustering was first used with the aim of improving the retrieval performance of existing IR systems.[4]

[3] Unfortunately and rather confusingly, the terms "clustering" and "classification" have occasionally also been used as substitutes (e. g. by van Rijsbergen, 1979).

[4] For an early overview of document clustering see Jardine and van Rijsbergen (1971).

Traditional IR systems compare a query with every document in the collection and return those that best meet certain similarity criteria. An alternative is *cluster-based retrieval*, which involves these differences:

1. Documents are clustered in an off-line process (in advance).
2. Queries are matched against clusters instead of documents.
3. One or several clusters are returned rather than individually chosen documents.

If implemented successfully, this procedure promises a number of advantages over an ordinary IR system:

- Faster execution because of a reduced number of comparisons for each query.
- Better *recall*[5] because relevant documents are still found even if they happen to use a slightly different terminology.
- Higher *precision*[6] because single documents that accidentally share many terms with the query but in terms of content belong to a different subject are not retrieved.

That being said, it must be noted that it is not at all obvious that the procedure can be made to work. As observed by van Rijsbergen, cluster-based retrieval can only work if the *cluster hypothesis* (Jardine and van Rijsbergen, 1971) holds:

> **Cluster Hypothesis.** *Closely associated documents tend to be relevant to the same requests* (van Rijsbergen, 1979, 45).[7]

Provided the hypothesis is true—i. e. if relevant and non-relevant documents for a specific query are well separated—the aforementioned improvements may be gained by statically clustering the document collection. However, van Rijsbergen (1979, 45–48) mentions two further important requirements:

- A clustering algorithm which is able to exploit the relevant associations among documents, which is stable with regard to new additions and small errors, which is independent of the order in which the documents are processed and which easily scales up.
- An algorithm to efficiently retrieve the best-matching cluster(s) for any given query.

In practice results have been rather mixed and the desired goals were often missed (Jardine and van Rijsbergen, 1971; van Rijsbergen and Sparck Jones, 1973; Voorhees, 1985, 1986; Willett, 1988; Hearst and Pedersen, 1996). Therefore, cluster-based retrieval is no longer in the focus of the document clustering community. A lack of distinct success and the difficulties transporting the concepts to modern large-scale Web search engines have led to the demise of this particular clustering application.

Other clustering approaches in the retrieval domain include *query clustering* to identify similar queries and answer them in a uniform way and *keyword clustering* as a technique of *query expansion* (Sparck Jones, 1971; Xu and Croft, 1996; Chang and Hsu, 1998).

[5] For a definition see Section 1.6 (Eq. 1.17).

[6] For a definition see Section 1.6 (Eq. 1.18).

[7] For a recent discussion of the cluster hypothesis see the work by Tombros. He claims that the cluster hypothesis is always true if an appropriate, query-dependent similarity measure is chosen (Tombros and van Rijsbergen, 2001; Tombros, 2002; Tombros *et al.*, 2002).

1.4.2 Supporting Presentation of IR Results

The application of IR technology to such a huge and diversified text collection as the World Wide Web (WWW) has given rise to a number of new issues. Storage, indexing, updating and retrieval all needed thorough re-thinking. The same was true (and still is) for the *presentational* aspects. New methods are needed to efficiently handle result sets comprising not just a few dozens but hundreds and thousands of documents.

This challenge has been tackled from various directions, but research is still far from completed. Much effort has gone into the devising of sophisticated ranking algorithms which aim to maximise the likelihood of returning highly relevant quality documents at the top of the list. Most prominent among these methods is Google's PageRank™ citation ranking (Page *et al.*, 1998).

Beyond the Ranked List

However, relatively often it is impossible to determine which documents are most relevant with regard to a specific user's request. Be it that the query is very short and allows multiple interpretations, be it that the query as a whole or individual parts thereof are inherently ambiguous. Joshi and Jiang (2001) differentiate between three possible reasons:

- *Polysemy*: search words have multiple meanings.
- *Phrases*: a phrase may have a meaning different from that of the individual words (e. g. "Sunday Times").
- *Term dependency*: words in a composite term may be depending on each other (e. g. "Digital Equipment Corporation").

One approach to resolve these ambiguities consists of a model for the individual user, thus taking into account the *context* of each query. An example of such an *Information Management Assistant* tracing the individual user's behaviour is WATSON (Budzik and Hammond, 1999). Another approach is to leave the ambiguities unresolved and refrain from making any presumptions on the user's intent. Instead of a list, the user is presented with a *clustered view* of the retrieved documents, a concept put forth by Willett (1985) and much explored ever since.

Compared to the common ordered results list, the clustered view approach comes with a number of advantages:

- Fewer assumptions are made on the user's intention.
- Within a few system interactions the user can reach a much larger number of documents than in a sequential list. Especially with cluster *hierarchies*, significant gains in time can be achieved (Leuski, 2001).
- Good cluster descriptions immediately give the user an overview of the main topics within the retrieved set of documents.
- The often difficult task of formulating concise queries loses in importance.[8] Instead, the query may remain fairly general and unspecific, with common navigational skills replacing the more demanding query formulation techniques (cf. Chang and Hsu, 1998).

[8] In fact, even without clustering interfaces available, most users prefer very short queries. Spink *et al.* (2001) report an average query length of 2.4 terms, with less than 5% of the queries featuring Boolean operators. Clustering would thus seem to be a highly desirable feature. (The study was based on 1997 data from the Excite search engine.)

In fact, using a clustering interface to a search engine combines the two major Web content access techniques: *searching* and *browsing* (Large *et al.*, 1999, 143). Back in 1994 Bowman *et al.* had predicted that a combination of the two techniques would be necessary to keep pace with the fast development of the Web. Much document clustering work in recent years has been explicitly devoted to clustering Web search results.[9]

Document clustering as a means of search result presentation (also known as "ad-hoc" or "ephemeral" clustering) is different from document clustering for retrieval purposes. It can be characterised as follows:

- Clustering is performed *dynamically* ("on-the-fly"). Time is thus at a high premium and fast algorithms of crucial importance.
- The clustering algorithm must be scalable and flexible, managing small result sets just as well as very large ones.
- Clustering must be able to deal with a very large variety of documents of different lengths, structures, languages, etc.
- The system must be able to come up with sensible, humanly understandable descriptions of the clusters (*cluster digests*). Furthermore, navigation and visualisation of the clusters must be user-friendly.
- Link-structure, HTML tags and other special features may be exploited for document representation and clustering.
- Unlike in a static environment, questions concerning updates and maintenance of the cluster structure are irrelevant.

In analogy to Jardine and van Rijsbergen's original cluster hypothesis for retrieval, a second cluster hypothesis for ephemeral clustering may be formulated:

> **Second Cluster Hypothesis.** *Documents in a collection can be grouped in such a way that significant benefits result for different users intending to review different specific parts or the whole of the document collection.*

Implicitly, this hypothesis is assumed to be true by most researchers in the field. However, there is valid reason to cast doubts, especially since studies have shown that users often have strong individual preferences and divergent views of what constitutes a "good" or "correct" clustering (Macskassy *et al.*, 1998). Moreover, the suitability of a particular clustering may depend on the search situation. With different types of searches having different types of pages as objects (Pirolli *et al.*, 1996) the same may be true of clustering.

Information Seeking Theory

In order to improve document clustering applications in IR, the *context* of the search situation (Vakkari *et al.*, 1997) needs closer attention, leading us into the field of *information seeking theory*.

Systematic research into the technical, practical, cognitive and emotional aspects of information seeking dates back to the 1960s (e. g. Taylor, 1968). But only in the 1990s did it become

[9] E. g. Zamir and Etzioni (1998, 1999); Zamir (1999); Maarek *et al.* (2000); He *et al.* (2001); Stefanowski and Weiss (2003).

apparent that for a quickly growing number of people the traditional information skills such as filtering, acquiring and storing information were no longer enough. The ability to recognise information needs in time, to express them and to satisfy them by actively *seeking* for appropriate information has become essential, both in professional and everyday life. Accordingly, the research into information seeking which had been rather neglected for a long time, started to attract new attention (Vakkari *et al.*, 1997, 7) and various attempts have been made to bring IR and information and library sciences closer together again (Vakkari, 1999).

Various *information seeking models* have been described in literature (Bates, 1989; Yang, 1997; Marchionini and Shneiderman, 1988; Marchionini, 1995) and particular attention has been paid to the *information seeking process* (Marchionini, 1989, 1992, 1995; Large *et al.*, 1999; Shneiderman *et al.*, 1997; Shneiderman, 1998).

A more specific branch of research deals with the various search types and behaviours that occur in practice (Ellis, 1989; Ellis and Haugan, 1997; Choo *et al.*, 2000a,b; Yang, 1997; Rosenfeld and Morville, 1998; Devlin and Burke, 1997; Saunders and Sheffield, 1998; Broder, 2002).

Rosenfeld and Morville, for instance, distinguish four general types of information search (Rosenfeld and Morville, 1998, 101–103):

Known-item searching. The user looks for a clearly defined bit of information. He knows that his question has a single, correct answer. Known-item searching may be regarded as the most "easy" concept, well supported by traditional IR instruments. Two typical known-item search tasks are *homepage finding* and *named-page finding* (Ogilvie and Callan, 2003).

Existence searching. The user looks for some fairly well-defined bit of information but he does not know whether it actually exists and he may have difficulties understanding and describing his need accurately.

Exploratory searching. The user has an open question or idea on which he wants to learn more. He may not know what exactly he wishes to find and there is unlikely to be a single, correct answer to his need.

Comprehensive searching (research). Just as exploratory searching, comprehensive searching has an open character. The difference is that the searcher wants to uncover *everything* related to the particular topic, not just a few useful pages.

For most searches of the first kind (known-item searches), a straightforward ranked list will do. But for existence and, in particular, exploratory searching a clustering interface offers new and perhaps more efficient access methods to a set of retrieved documents. Finally, also a comprehensive search may profit from the order gained by the presentation in clusters.

Further systematic research is needed to combine the insights from information seeking theory with document clustering and other sophisticated IR techniques. Intuitively, however, a clustered view of search results has a number of promising applications.

1.4.3 Direct Access to Document Collections

Document clustering has further been employed in several innovative collection access methods which are not directly relying on text retrieval. Usually they are based on a combination of search and browse activities. Some of these methods are briefly described below. In nature they are often quite similar to the approaches discussed in the previous two sections.

Korfhage (1991) presents a system which, in principle, can display all documents of a collection in relation to a few user-defined *reference points*, thus allowing a situation-dependent navigational access. For obvious reasons, the system fails to scale up to Web standards.

Scatter/Gather (Cutting *et al.*, 1992, 1993; Hearst and Pedersen, 1996) is an algorithm which works on a result set or a pre-clustered collection. Rather than calculating a detailed cluster hierarchy, the crude top-level clustering is only further refined as need arises, i. e. as the user navigates his way up and down the hierarchy.

WebACE (Boley *et al.*, 1999a) is a system monitoring a user's activity on the Web and building a user profile by clustering frequently visited pages. In a second phase the system supports the user by presenting him with new documents found on the Web based on that user profile.

WebCluster (Harper *et al.*, 1999; Muresan, 2002) is a system offering *mediated access* to a potentially very large collection (such as the Web) by presenting users with a static clustered view of a selected, domain-specific sub-collection. The actual search is only launched after the user has identified the clusters that appear most pertinent to his information need.

1.5 Systems in Practice

Here follows a non-exhaustive list of systems that implement a more or less sophisticated clustering of documents and which can (or could) be found on-line on the Internet:

Grouper. A research project at Washington University using an innovative algorithm called *Suffix Tree Clustering* (Zamir and Etzioni, 1999; Zamir, 1999). Since the year 2000 the clustering interface *Grouper* and the underlying meta-search engine *HuskySearch* are no longer publicly available.

NorthernLight.[10] This used to be a Web search engine which displayed its search results grouped into a number of predefined classes. Thus, it does not actually use clustering but a classification algorithm. After a change in proprietorship the company is specialising in enterprise search solutions which also include a clustering algorithm. It no longer provides a free Web search service, however.

WiseNut.[11] A Web search engine powered by *LookSmart* which groups results into a varying number of clusters. The algorithm seems to work exclusively with page titles.

Teoma.[12] A Web engine with a standard list of most relevant documents, but which offers in a sidebar a number of rudimentary refinement options derived from a clustering of the Web pages into so-called "communities".

KartOO.[13] A commercial content search solution with a highly advanced and dynamic graphic interface for displaying contents and documents, also including clustering functionalities.

Vivísimo.[14] A commercial content clustering solution, coming with a popular and free meta-search engine as an illustration of its document clustering capabilities. Its special feature is hierarchical *conceptual* clustering, resulting in clusters which it is claimed can be described "concisely, accurately, and distinctively" (Vivísimo, 2003). A number of raving white papers at their Web site propagate the many virtues of clustering technology.

[10] `www.northernlight.com`
[11] `www.wisenut.com`
[12] `www.teoma.com`
[13] `www.kartoo.net`
[14] `www.vivisimo.com`

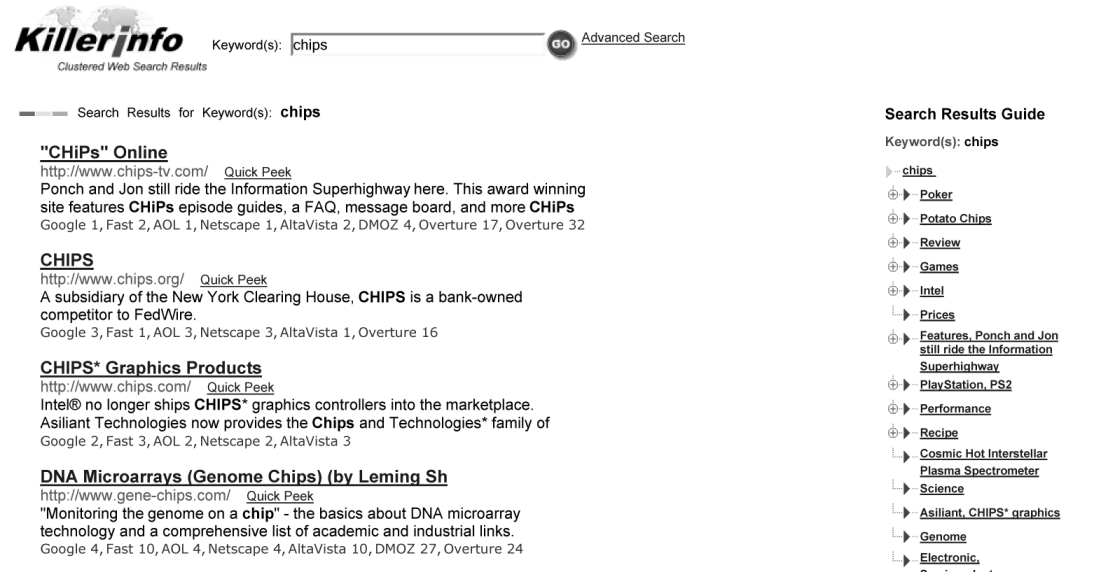


Figure 1.3: **Killerinfo.** A view of the results page of the Killerinfo meta search engine, with clusters in the right half of the window.

Search Allinone MetaSeach.[15] Another, more recent meta search engine which presents a twofold view of the results: ordered list and grouped results.

Killerinfo.[16] A meta search engine with considerable stress on the hierarchically clustered presentation of search results. Figure 1.3 shows the Killerinfo interface.

Entrieva.[17] A company developing various professional knowledge management tools which use, among others, taxonomies and algorithms for automatical taxonomy generation.

1.6 Notation

This section introduces the formal notation used in the following chapters. It refers to a standard document clustering application within a standard IR system. Special systems may require notations and definitions differing from those given here.

For simplicity's sake and in accordance with the majority of the relevant literature, the distinction between row and column vectors is only made explicit where necessary. In those cases, $\mathbf{x}$ refers to a column vector and $\mathbf{x}^T$ to a row vector.

Document. Let D_i abstractly denote an individual document i.

[15] `www.searchallinone.com`
[16] `www.killerinfo.com`
[17] `www.entrieva.com`

Document Universe. Let Ω be the *document universe*, i. e. the set of all documents known to the IR system:

$$\Omega = \{D_i \mid i \in \mathbf{N}\}. \tag{1.1}$$

$\mathbf{N}$ is the set of natural numbers.

Document Set. Let $\mathcal{S}$ be the *document set*, i. e. those n documents that form the input to the clustering process:

$$\mathcal{S} = \{D_1, D_2, \ldots, D_n\} \subseteq \Omega. \tag{1.2}$$

Note: For many—but not all—applications $\mathcal{S}$ equals Ω.

Feature Set. Let

$$\mathcal{F} = \{f_1, f_2, \ldots, f_m\}, \tag{1.3}$$

with $\mathcal{F}$ a *set of m features* and f_i an individual feature i. Each feature stands for a concrete or abstract document property.

Document Vector. Let

$$\mathbf{d}_i = (d_{i1}, d_{i2}, \ldots, d_{im}), \tag{1.4}$$

with $\mathbf{d}_i$ the *document vector* of document D_i in an m-dimensional feature space $\mathcal{F}$. The jth component of $\mathbf{d}_i$ (written as d_{ij}) corresponds to the *value* or *strength* of feature f_j in document D_i. d_{ij} is usually a non-negative real number:

$$d_{ij} \in \mathbf{R}_0^+. \tag{1.5}$$

Document Feature Matrix. Let

$$H = \begin{pmatrix} \mathbf{d_1} \\ \mathbf{d_2} \\ \vdots \\ \mathbf{d_n} \end{pmatrix}, \tag{1.6}$$

with H the *document feature matrix*, defined by the individual vector representations $\mathbf{d}$ of all $D \in \mathcal{S}$. The document feature matrix is usually the input to the clustering algorithm.

Document Vectorisation Function. Let τ be a function which transforms a text document into an m-dimensional vector representation in feature space $\mathcal{F}$:

$$\mathbf{d}_i = \tau(D_i), \text{ with } \tau : \Omega \to \mathbf{R}^m, \tag{1.7}$$

with $\mathbf{R}$ the set of real numbers.

Feature Transformation Function. Let ϕ be a function which transforms a document vector from one feature space ($\mathcal{F}_1$) into another ($\mathcal{F}_2$), sometimes making use of additional information from the document feature matrix H:

$$\mathbf{d}'_i = \phi(\mathbf{d}_i, H), \text{ with } \phi : \mathbf{R}^{m_{\mathcal{F}_1}}, \mathbf{R}^{n \times m} \to \mathbf{R}^{m_{\mathcal{F}_2}}. \tag{1.8}$$

Document Frequency. Let

$$\mathrm{df}(j, H) = \sum_i |\mathrm{sgn}(h_{ij})|, \tag{1.9}$$

with the *document frequency* $\mathrm{df}(j, H)$ the number of documents with a non-zero value for feature f_j.

Cluster. Let a *cluster* C_i be an subset of $\mathcal{S}$:

$$C_i \subseteq \mathcal{S}, \tag{1.10}$$

and let n_i be the number of objects in cluster C_i:

$$n_i = |C_i|. \tag{1.11}$$

Cluster Solution. Let

$$\mathcal{C} = \{C_1, C_2, \ldots, C_k \mid C_i \subseteq \mathcal{S} \;\; \forall i \in 1 \ldots k\}. \tag{1.12}$$

A *cluster solution* $\mathcal{C}$ is thus defined as a set of k clusters.

Cluster Algorithm. Let

$$\mathcal{C} = \kappa(H), \quad \text{with} \quad \kappa : \mathbf{R}^{n \times m} \rightarrow \mathcal{P}(\mathcal{S}) \tag{1.13}$$

and with κ denoting the cluster algorithm, $\mathcal{P}(\mathcal{S})$ the power set of $\mathcal{S}$ and $\mathbf{R}$ the set of real numbers.

Cluster Representative. Let

$$\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{im}), \tag{1.14}$$

with $\mathbf{r}_i$ a *representative vector* for cluster C_i in an m-dimensional feature space $\mathcal{F}$.

Individual Cluster Criterion Function. An individual *criterion function* $E(C)$ measures the quality of a single cluster:

$$E : \mathcal{P}(\mathcal{S}) \rightarrow \mathbf{R}, \tag{1.15}$$

with $\mathcal{P}(X)$ the power set of X and $\mathbf{R}$ the set of real numbers.

Overall Cluster Criterion Function. An overall *criterion function* $\Psi(\mathcal{C})$ measures the quality of an entire cluster solution:

$$\Psi : \mathcal{P}(\mathcal{P}(\mathcal{S})) \rightarrow \mathbf{R}, \tag{1.16}$$

with $\mathcal{P}(\mathcal{P}(\mathcal{S}))$ the set of all possible cluster solutions.

Type and Token. Within documents it is common to refer to word *types* and word *tokens*. The former refer abstractly to features in a document or a corpus, while the latter refer to individual occurrences. Formally speaking, the tokens of a document are a *bag* (which allows multiple occurrences of the same element). The types are the *set* created by eliminating all duplicates from the token bag.

Recall and Precision. In IR two widespread performance measures are defined by the set of documents in a collection that are *relevant* to a particular query ($\mathcal{A}$) and those documents that are actually *retrieved* by the system ($\mathcal{B}$):

$$\textit{Recall}\ (R) \quad = \quad \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|}, \tag{1.17}$$

$$\textit{Precision}\ (P) \quad = \quad \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{B}|}\ . \tag{1.18}$$

Their weighted arithmetic mean, the so-called F-Measure is also used frequently (see Equation 2.77 for an example).

For guidance to the notation used in the experimental results tables, please consult Section 4.3.1.

Chapter 2

Cluster Analysis

For instance, bird differs from bird
by gradation, or by excess and defect;
some birds have long feathers,
others short ones, but all are feathered.
Bird and Fish are more remote
and only agree in having analogous organs;
for what in the bird is feather,
in the fish is scale.

Aristotle (*On the Parts of Animals*)

For millennia grouping similar natural phenomena together and classifying them into categories has been a fundamental and vital strategy for mankind to find its way in a complex and dangerous environment. It is thus not surprising that the first systematic classification schemes for animals date back to Aristotle (384–322 BC) and his pupil Theophrastos ($c371$–$c287$ BC). The urge to bring order into things by creating clusters and classification schemes has grown ever since.

But even though evolution has shaped the human mind into a classifier *par excellence*, the information age has also shown us our limits: although we have no problem dealing with small-sized data sets in 2-dimensional spaces, our mind is ill-equipped for dealing with thousands and millions of data points and unstructured high-dimensional feature spaces. Here systematic *cluster analysis* must set in, which in turn may provide the foundations for a systematic classification scheme and later serve as a basis for an automated *categorising* procedure.

Cluster analysis has almost an infinite number of applications in a wide spectrum of sciences such as archaeology, astronomy, biology, chemistry, medicine, market research, pattern recognition, psychiatry and many others (Duda and Hart, 1973; Everitt, 1993; Berry and Linoff, 1997). In particular *biological taxonomy* has long been a driving force in the development of powerful and sophisticated algorithms (Sneath and Sokal, 1973). These in turn have boosted research and application in the other areas as well.

Clusters

Clustering is a process aimed at finding clusters in data. But what exactly *is* a cluster? Practice has shown this to be a surprisingly difficult question to answer. Jain and Dubes (1988, 1)

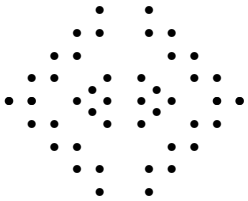


Figure 2.1: **What is a cluster?** Situations may arise where no cluster solution is universally "correct". In this example different solutions are equally plausible.

define a cluster as a "number of *similar* objects collected or grouped together". Capturing the phenomenon more formally is not easy and may even be misguiding (Everitt, 1993, 6). Everitt mentions further definition attempts, including some based on internal cohesion and external isolation of individual clusters.

Why too rigid a definition of "cluster" may not be desirable is illustrated by Figure 2.1. How many clusters can be discerned? And if the answer shall be two, which data points belong together?

It becomes evident that despite the great efforts to get a tight formal grip on clustering, a task-inherent arbitrariness will always remain and for many situations a single "correct" criterion for defining the clusters cannot be guaranteed to exist. Depending on the context and individual preferences, different solutions may suit best.

This insight has been confirmed for document clustering by a study on human performance in clustering Web pages. It was shown that the participants developed distinctly different cluster solutions for the same set of documents (Macskassy *et al.*, 1998). Especially in a high-dimensional domain such as document clustering the optimal solution is thus often subjective to some degree.

The rest of this chapter seeks to provide an overview of the more important formal aspects of cluster analysis. Sections 2.1 to 2.4 deal with data structures, similarity measures and clustering algorithms. Sections 2.5 and 2.6 deal with cluster properties and evaluation/validation methods.

Individual clustering techniques and terminology vary from one application area to another. The present introduction will focus on those aspects relevant for the *document* clustering domain.[1]

2.1 Data Representation

In order to automatically cluster real-world phenomena (objects), they need to be captured and translated into formal representations. We distinguish between three principal possibilities of viewing the data: the vector space model, the metric space model and the graph model.

2.1.1 Vector Space Model

In the vector space model, each object is represented by an m-dimensional vector ($\mathbf{d}_i$). Each dimension represents a certain property $f_j \in \mathcal{F}$ found in the data while the individual values d_{ij} indicate the presence and/or extension of that particular property f_j for the given object i. Absence is usually indicated by zeroes.

[1] For a general in-depth introduction to cluster analysis see Jain and Dubes (1988).

It is the task of the system designer to identify a suitable feature space $\mathcal{F}$ and providing a mapping τ from the real-world objects into this feature space (which will form the topic of Chapter 3). The features can have binary, nominal, ordinal or numeric values; some models even allow a mix of different scales.

For document clustering it is customary for the vectors to consist of real, non-negative numbers ($d_{ij} \in \mathbf{R} \geq 0$).

2.1.2 Metric Space Model

Under certain circumstances, the vector space model is not applicable—for instance if the data is (partially) non-numeric or if the base data is missing. In such a case it may still be possible to represent the objects formally, if we have enough information about the relationship of the objects among themselves in the form of a $n \times n$ *similarity* or *distance* matrix.

By applying a similarity measure it is always possible to transform an $n \times m$ vector space model into a $n \times n$ metric space model but not vice versa. For a discussion of algorithms taking account of the restrictions of metric space models see Baeza-Yates *et al.* (2003) and for ROCK, an algorithm suitable for categorical data, see Guha *et al.* (2003).

2.1.3 Graph Model

The vast majority of clustering algorithms and applications use the vector or metric space model, but there exists an alternative in the *graph-based* view of the document space:

- *Graph partitioning* as a means of data/document clustering has enjoyed notable popularity (see Ding *et al.*, 2001; He *et al.*, 2001). Here the data is represented in an undirected weighted graph $G = (V, E, W)$ where the vertices V correspond to the individual objects and the edges E to the associations between the objects, while W is a non-negative weighting scheme for these associations (often based on similarity measures such as those in Section 2.2). For document clustering a completely connected graph is often chosen, but alternative graphs (e. g. inspired by hyper-link structures among the documents) also come into consideration.

  The clustering task is to split the graph into disjoint sub-graphs (partitions) following a certain objective function. In the simplest case (MINcut) the objective function is to minimise the *cut size*. Cut size is defined as the sum of the weights of all edges that need to be cut to separate the partitions.[2] Since this method often leads to very skewed partitions, various improvements have been suggested such as *ratio cut*, *normalised cut* and *Min-max cut* (Ding *et al.*, 2001). The latter, for instance, aims simultaneously to minimise cut size and maximise the weights within a partition.[3]

  Finding optimal solutions to the graph partitioning problem is NP-complete, but graph theory being a well-understood subject, a number of powerful approximation methods are available.

[2]
$$Cut(A, B) = \sum_{i \in A, j \in B} W_{ij}, \text{ with } A, B \subset V \text{ and } A \cap B = \emptyset. \tag{2.1}$$

[3]
$$Mcut = \frac{Cut(A, B)}{W(A)} + \frac{Cut(A, B)}{W(B)}, \text{with } W(X) = \sum_{i,j \in X} W_{ij}. \tag{2.2}$$

- *Association Rule Hypergraph Partitioning* is a related approach. A hypergraph differs from a common graph in that an edge does not connect just two but arbitrarily many vertices. After discovering *association rules* (Agrawal *et al.*, 1993) between data points, so-called *frequent itemsets* are built—overlapping sets of objects sharing certain distinctive features. After projecting these frequent itemsets onto *hyperedges* between the objects, similar graph partitioning methods as above can be applied (Moore *et al.*, 1997; Boley *et al.*, 1999a,b; Clifton *et al.*, 2004; Noel *et al.*, 2003).

- *Co-clustering* is another form of graph-based clustering. It involves a bi-partite graph to represent objects (documents) on the one side and features (words) on the other. A special algorithm then clusters both sides *simultaneously* (Dhillon, 2001).

In the following discussion graph and metric space models will be left aside and we will focus on the wide-spread vector space model.

2.2 Similarity Measures

Clustering being the process of identifying groups of objects that are similar to each other, a *measure of similarity/dissimilarity* between two objects lies at the core of all clustering algorithms. Virtually all of them are based on an explicit definition of such a measure in the vector space, and many algorithms in fact take as input an $n \times n$ similarity (proximity) matrix rather than the bare $n \times m$ object-feature matrix.

The present section gives a short overview of the different similarity coefficients encountered in document clustering practice. For a more detailed and formal discussion of similarity coefficients refer to Sneath and Sokal (1973) and Jain and Dubes (1988).

Section 2.2.1 will deal with similarity between two individual objects and Section 2.2.2 with the problem of measuring similarity between entire groups of objects. In each case the discussion relies on the vector space model.

2.2.1 Inter-Object Similarity

Measures for determining the similarity between two vectors range from simple geometrical to sophisticated statistical methods. An ordinary similarity measure s assumes values from 0 (complete dissimilarity) to 1 (total identity).

Metric Distance Coefficients

The most intuitive similarity coefficients measure the *distance* between two points in a vector space and will be denoted by $\hat{s}$.[4]

Usually, distance metrics have the form of a Minkowski metric:

$$\hat{s}(\mathbf{d}_i, \mathbf{d}_k) = \left[\sum_{j=1}^{m} |d_{ij} - d_{kj}|^r \right]^{1/r} \quad \text{where } r \geq 1. \tag{2.3}$$

The three most common parameters are

[4] A *distance* metric $\hat{s}$ can always be easily transformed into an ordinary $[0, 1]$ similarity measure s, e. g. with $s = \frac{1}{1+\hat{s}}$ or $s = e^{-\hat{s}^2}$ (Strehl *et al.*, 2000).

- $r = 2$, the well-known *Euclidean distance* (also known as the L_2 norm):

$$\hat{s}_{\text{Euclid}}(\mathbf{d}_i, \mathbf{d}_k) = \sqrt{\sum_{j=1}^{m} (d_{ij} - d_{kj})^2} = \|\mathbf{d}_i - \mathbf{d}_k\|_2 , \tag{2.4}$$

- $r = 1$, the *Manhattan* or city block distance:

$$\hat{s}_{\text{Manhattan}}(\mathbf{d}_i, \mathbf{d}_k) = \sum_{j=1}^{m} |d_{ij} - d_{kj}| = \|\mathbf{d}_i - \mathbf{d}_k\|_1 , \tag{2.5}$$

- $r \to \infty$, the *Chebyshev* distance:

$$\hat{s}_{\text{sup}}(\mathbf{d}_i, \mathbf{d}_k) = \max_{1 \le j \le m} |d_{ij} - d_{kj}| = \|\mathbf{d}_i - \mathbf{d}_k\|_\infty . \tag{2.6}$$

Most often preference is given to the Euclidean metric.

Association Coefficients

Association coefficients aim to measure the *agreement* between two vectors in the individual positions. Among the numerous formulae that have been suggested the *Dice Coefficient*

$$s_{\text{Dice}}(\mathbf{d}_i, \mathbf{d}_k) = 2 \sum_{j=1}^{m} d_{ij} d_{kj} \Big/ \left(\sum_{j=1}^{m} {d_{ij}}^2 + \sum_{j=1}^{m} {d_{kj}}^2 \right) \tag{2.7}$$

and the (extended) *Jaccard Coefficient*

$$s_{\text{Jaccard}}(\mathbf{d}_i, \mathbf{d}_k) = \sum_{j=1}^{m} d_{ij} d_{kj} \Big/ \left(\sum_{j=1}^{m} {d_{ij}}^2 + \sum_{j=1}^{m} {d_{kj}}^2 - \sum_{j=1}^{m} d_{ij} d_{kj} \right) \tag{2.8}$$

enjoy the greatest popularity.

Both coefficients measure the commonness of features between two objects, divided by a normalisation coefficient. Initially, these measures had been developed for binary data; the above are generalised forms for non-binary data as are typically found in the information retrieval domain. Several very similar variations exist of these two coefficients (cf. Salton and McGill, 1983; Strehl *et al.*, 2000; Hammouda, 2001).

Another association coefficient is the *MinMax* distance used in the experiments of Bellot and El-Bèze (2000):

$$\hat{s}_{\text{MinMax}}(\mathbf{d}_i, \mathbf{d}_k) = 1 - \frac{\sum_{j=1}^{m} \min(d_{ij}, d_{kj})}{\max\left(\sum_{j=1}^{m} d_{ij}, \sum_{j=1}^{m} d_{kj} \right)} . \tag{2.9}$$

Cosine similarity. For document clustering, however, the most popular coefficient has been *cosine similarity*, which measures the *angle* between two document vectors (Salton and McGill, 1983) and thus ignores any differences in length:

$$
\begin{aligned}
s_{\text{Cosine}}(\mathbf{d}_i, \mathbf{d}_k) &= \frac{\mathbf{d}_i^T \mathbf{d}_k}{\|\mathbf{d}_i\|_2 \cdot \|\mathbf{d}_k\|_2} \\
&= \sum_{j=1}^{m} d_{ij} d_{kj} \Big/ \sqrt{\sum_{j=1}^{m} {d_{ij}}^2 \cdot \sum_{j=1}^{m} {d_{kj}}^2} \quad . \qquad (2.10)
\end{aligned}
$$

If the vectors are normalised to unit length (as is often done in document clustering, see Section 3.3.3), it can be shown that cosine similarity is equivalent to the Euclidean distance metric. In fact, the following relation holds (see the Appendix for the proof, Section B.1):

$$
\begin{aligned}
\hat{s}_{\text{Euclid}}(\mathbf{d}_i, \mathbf{d}_j) &= \sqrt{2 - 2 s_{Cosine}(\mathbf{d}_i, \mathbf{d}_j)}, \\
&\quad \text{if } \|\mathbf{d}_i\|_2 = \|\mathbf{d}_j\|_2 = 1. \qquad (2.11)
\end{aligned}
$$

Cosine similarity is efficient to compute and having repeatedly produced reliable results for document clustering, it is the common choice in most IR clustering systems.[5]

Statistical Coefficients

More sophisticated similarity measures with a sound statistical foundation exist, including the *Pearson correlation coefficient* (cf. Sneath and Sokal, 1973):

$$
s_{\text{Pearson}}(\mathbf{d}_i, \mathbf{d}_k) = \frac{1}{2} \left(\frac{(\mathbf{d}_i - \bar{d}_i)(\mathbf{d}_k - \bar{d}_k)}{\|\mathbf{d}_i - \bar{d}_i\|_2 \, \|\mathbf{d}_k - \bar{d}_k\|_2} + 1 \right), \qquad (2.12)
$$

where $\bar{d}_i$ is the average feature value of $\mathbf{d}_i$ over all dimensions ($\bar{d}_i = \sum_{j=1}^{m} d_{ij}/m$).

The significance of the computationally demanding correlation coefficients for document clustering is small. A study by Strehl *et al.* (2000) compared Pearson correlation with Euclidean, cosine, and extended Jaccard coefficients. Their result indicates that cosine and Jaccard coefficients are the most useful for clustering, with the Pearson coefficient not too far behind and Euclidean distance performing poorly.

A *probabilistic approach*, measuring the expected overlap between two documents under a particular "corpus model", is presented by Goldszmidt and Sahami (1998). In their experiments the probabilistic measure performed favourably compared to cosine similarity, which they show to be a special case of probabilistic overlap.

2.2.2 Inter-Cluster Similarity

A substantial number of clustering algorithms (notably the hierarchical agglomerative algorithms discussed in Section 2.4.1) require similarity computations not just among individual objects but also among entire clusters (with the single-object cluster being just a special case).

Often the comparison is reduced to the comparison between two individual vectors that "represent" the two clusters. Different methods for determining those "representatives" may lead

[5] Tombros and van Rijsbergen (2001), working on an ad-hoc clustering system for document retrieval, propose to extend cosine similarity by a bias towards terms that were present in the initial user query. In their model, overlap in *query* terms is rewarded by an extra addition to the similarity score. They report significant improvements over plain cosine measures. However, for obvious reasons the effect of their method is much greater for queries with many "OR"-connected terms than for the more typical queries consisting of a few "AND"-connected terms where by definition each retrieved document must contain all the query terms.

to very different results. The choice of method therefore often reflects the designer's assumptions/wishes about the clusters to be found.

In the following discussion vector ${}_{[j]}\mathbf{r}_i^{\mathrm{x}}$ denotes the vector representing cluster C_i and x indicates the method used. If the choice of the cluster representative depends on the second cluster under consideration, then the optional parameter j is introduced to indicate the second cluster. Unless stated otherwise, it is assumed that all clusters are disjoint ($C_1 \cap C_2 = \emptyset$).

Given two clusters C_1 and C_2, these methods are known to compute a measure of similarity $S(C_1, C_2)$ between them (cf. also Sneath and Sokal, 1973; Jain and Dubes, 1988):

Centroid. Each cluster is represented by an averaged version (the mean) of all its constituents:

$$\begin{aligned} S(C_1, C_2) &= s(\mathbf{r}_1^{\mathrm{c}}, \mathbf{r}_2^{\mathrm{c}}), \\ \text{with} \quad \mathbf{r}_i^{\mathrm{c}} &= \frac{\sum \mathbf{d}_j}{n_i},\ \mathbf{d}_j \in C_i. \end{aligned} \tag{2.13}$$

Sometimes centroids are normalised to have unit length:

$$\mathbf{r}_i^{\hat{\mathrm{c}}} = \frac{\sum \mathbf{d}_j}{\|\sum \mathbf{d}_j\|_2},\ \mathbf{d}_j \in C_i. \tag{2.14}$$

Schütze and Silverstein (1997) work with truncated centroids but they have not had many followers.

Centroids are usually straightforward to calculate and maintain.

Medoid. The centroid being an artificial and sometimes unnatural choice, a "most central" member of the cluster, a *medoid*, is occasionally chosen instead. The medoid can be defined as follows:

$$\begin{aligned} S(C_1, C_2) &= s(\mathbf{r}_1^{\mathrm{m}}, \mathbf{r}_2^{\mathrm{m}}), \\ \text{with} \quad \mathbf{r}_i^{m} &= \arg\max_{\mathbf{d}_j \in C_i} s(\mathbf{d}_j, \mathbf{r}_i^{\mathrm{c}}). \end{aligned} \tag{2.15}$$

One of the medoid's advantages is the reduced influence of outliers, but it also requires more time to determine than the centroid.

Medoids have been used in various clustering algorithms such as *PAM*[6], *CLARA*[7] and *CLARANS*[8]. See Kaufman and Rousseeuw (1990) and Chu *et al.* (2002) for an overview of medoid-based algorithms.

Nearest-Neighbour. Nearest-neighbour representation (also known as the *single-linkage* method) chooses for each pair of clusters the nearest two vectors:

$$S(C_1, C_2) = \max s(\mathbf{d}_a, \mathbf{d}_b), \quad \text{with} \quad \mathbf{d}_a \in C_1, \mathbf{d}_b \in C_2. \tag{2.16}$$

[6] *PAM* is an acronym for *Partitioning Around Medoids*.
[7] *CLARA* is an acronym for *Clustering LARge Applications*.
[8] *CLARANS* is an acronym for *Clustering Large Applications based on RANge Search*.

The representative vector thus depends on the second cluster (C_j) and may be different for every comparison:

$$ {}_j\mathbf{r}_i^{\text{nn}} = \arg\max_{\mathbf{d}_a \in C_i} s(\mathbf{d}_a, \mathbf{d}_b), \text{ with } \mathbf{d}_b \in C_j. \tag{2.17} $$

Nearest-neighbour comparisons thus always take the "best" match. All the other cluster members have no influence.

Furthest-Neighbour. The opposite of nearest neighbour is *furthest neighbour* (*complete linkage*), which uses the most distant member:

$$ S(C_1, C_2) = \min s(\mathbf{d}_a, \mathbf{d}_b) \quad \text{with} \quad \mathbf{d}_a \in C_1, \mathbf{d}_b \in C_2. \tag{2.18} $$

and

$$ {}_j\mathbf{r}_i^{\text{fn}} = \arg\min_{\mathbf{d}_a \in C_i} s(\mathbf{d}_a, \mathbf{d}_b), \text{ with } \mathbf{d}_b \in C_j. \tag{2.19} $$

All members are thus taken into account ("complete linkage") and the "worst" match is chosen.

Group Average. The group average method does not use an individual vector to represent a cluster. Instead the arithmetic mean of all individual vector comparisons between the two clusters is taken.

$$ S(C_1, C_2) = \frac{1}{n_1 \cdot n_2} \sum_{\mathbf{d}_a \in C_1, \mathbf{d}_b \in C_2} s(\mathbf{d}_a, \mathbf{d}_b). \tag{2.20} $$

Minimum Variance. The minimum variance method generates a distance measure between two groups by measuring the *information loss* (= error sum of squares = ESS) incurred by their merging into one group. The smaller the information loss, the more similar are the two groups:

$$ \begin{aligned} \hat{S}(C_1, C_2) &= \text{ESS}(C_1 \cup C_2) - \text{ESS}(C_1) - \text{ESS}(C_2), \\ \text{with} \quad \text{ESS}(C_i) &= \sum_{\mathbf{d}_j \in C_i} \|\mathbf{d}_j - \mathbf{r}_i^{\text{c}}\|_2^2 . \end{aligned} \tag{2.21} $$

As $\text{ESS}(C_i) = n_i V(C_i)$, the same can be expressed in terms of variance V:

$$ \begin{aligned} \hat{S}(C_1, C_2) &= (n_1 + n_2)V(C_1 \cup C_2) - n_1 V(C_1) - n_2 V(C_2), \\ \text{with} \quad V(C_i) &= \frac{1}{n_i} \sum_{\mathbf{d}_j \in C_i} (\mathbf{d}_j - \mathbf{r}_i^{\text{c}})^T (\mathbf{d}_j - \mathbf{r}_i^{\text{c}}) . \end{aligned} \tag{2.22} $$

Equation 2.21 resp. 2.22 can be further simplified to[9]

$$ \hat{S}(C_1, C_2) = \frac{(\|\mathbf{r}_1^c - \mathbf{r}_2^c\|_2)^2}{\frac{1}{n_1} + \frac{1}{n_2}} . \tag{2.23} $$

[9] Cf. Appendix B.2.

Other Cluster Representative Selection Models. Clustering with the aforementioned methods is often suitable for some distributions but unsuitable for others. For instance, *furthest neighbour* is good at recognising small, tight clusters, but bad at dealing with outliers and loose clusters, while *nearest neighbour* often results in undesirably long and "thin" clusters, and minimum variance has a marked tendency towards producing clusters of equal sizes.

This prompted the development of more sophisticated cluster comparison methods:

- One such method is implemented in *CURE*[10] (Guha *et al.*, 1998, 2003). The method works on the *nearest-neighbour* principle, but instead of all the members of a cluster it uses a specific set of "representatives". These are a constant number of data points for each cluster and calculated as follows: the first representative is the point furthest away from the centroid of the cluster; the second representative is the cluster point furthest away from the first representative; the third is the one furthest away from the second; and so on, until the desired number r of representatives is found. Before single-linkage sets in, all representatives are shrunk towards the cluster centroid by a constant factor $\alpha \in [0, 1]$. It has been shown that CURE is able to detect clusters of unusual forms that are not identified by other measures.
- In the domain of document clustering a related idea has been put forth by Bellot and El-Bèze (2000). As in the medoid approach, their goal is to work with real documents only. Instead of a single medoid, however, they use the r documents nearest to the centroid. Comparisons between these representatives and/or individual documents are again based on the single-linkage principle.
- Cutting *et al.* (1992) use a centroid approach, but rather than computing the centroid of all documents they suggest using a so-called "trimmed profile" (a sort of centroid), which cancels out the effects of outliers. The trimmed profile is computed from the r documents nearest to the "untrimmed" centroid (r being either a constant or a fraction of the number of documents in the cluster).

Other approaches to measuring similarity exist, for example those based on the *Mahalanobis* distance, taking the within-cluster correlations of individual variables (terms) also into account (cf. Everitt, 1993) or those based on the number of "shared nearest neighbours" (cf. Ertöz *et al.*, 2003).

2.3 Non-Hierarchical Algorithms

In theory, cluster analysis is absolutely trivial. Three simple steps suffice (see Figure 2.2).

In practice, however, this approach turns out to be unfeasible, for the number of possible cluster solutions increases dramatically with the number of objects and clusters. For n objects and k non-hierarchical, non-overlapping clusters the number of possible cluster solutions is approximatively $k^n/k!$ (Duda and Hart, 1973, 226), which means that for the modest number of 50 objects and 5 clusters already over 10^{32} cluster solutions would have to be checked.

For most real applications, exhaustive search is therefore out of the question and heuristic optimisation methods are compulsory. Cluster algorithms come in an abundance of flavours. Generally, they can be divided in two large classes: *hierarchical* and *non-hierarchical* algorithms.

[10] *CURE* is an acronym for *Clustering Using REpresentatives*.

1. Select a suitable *clustering criterion function* $\Psi(\mathcal{C})$, also known as the *objective function*. It is a measure of quality of cluster solution $\mathcal{C}$.
2. Enumerate all possible cluster solutions $\mathcal{C}_1, \ldots, \mathcal{C}_z$ and compute the value of the objective function for each solution.
3. Pick the solution with the highest value.

Figure 2.2: **Trivial clustering algorithm.** In theory cluster analysis is a trivial task which can be solved by three simple steps.

- *Hierarchical algorithms* produce not just a bunch of clusters but by either merging or dividing clusters they create entire hierarchies, similar to those known from biological taxonomy. They are dealt with in Section 2.4.
- *Non-hierarchical algorithms* work mostly on the hill-climbing principle and produce a flat selection of clusters. Their expressiveness is therefore smaller than that of the hierarchical algorithms. Non-hierarchical algorithms are dealt with in the rest of this section.

2.3.1 Iterative Partitional Clustering

The most typical non-hierarchical clustering algorithms are the *partitional iterative* algorithms. They come with a complexity order of $O(n)$ or $O(n \log n)$ (Willett, 1988) and are often used to good effect in reasonable time, with many successful applications in document clustering.

The standard forms of iterative partitional clustering are dealt with in the present section, while Section 2.3.2 gives a brief overview of some alternatives and deviations.

A partitional iterative algorithm is characterised by

- a series of *iterations* or repetitions by which the cluster solution is incrementally optimised with regard to a certain cluster criterion;
- the objects being divided in real *partitions*, i. e. all objects belong to exactly one cluster, with is no overlap between clusters and no superimposed structure (such as a hierarchy).

Most iterative algorithms are non-deterministic, i. e. different initial seeds may lead to entirely different solutions because the hill-climbing technique is *locally* looking for an optimum.

An iterative clustering process involves the five principal steps shown in Figure 2.3 (cf. Modha and Spangler, 2003; Everitt, 1993, Chapter 5). Identifying the suitable number of clusters k in step 2 is often crucial and further discussed in Sections 2.5.3 and 2.6.4. The other steps are one by one dealt with in what follows.

2.3.1.1 Cluster Criterion Function

We distinguish local and global cluster criteria.

Local criteria work by simply determining the nearest cluster for each document. The algorithm thus keeps moving documents to their nearest clusters. Since the clusters change in the process, the problem is not solved in a single step and many iterations may be necessary until a stable solution is found. The nearest cluster can be found by using the similarity measures outlined in Section 2.2.2.

1. Decide upon a cluster criterion.
2. Select an initial (random) partition with k clusters.
3. With regard to the cluster criterion evaluate all possible moves of individual documents from one cluster to another.
4. Perform those move(s) that actually improve the cluster solution.
5. Repeat steps 3 and 4 until a stop criterion is met.

Figure 2.3: **Iterative clustering algorithm.** The five principal steps of an iterative clustering algorithm.

Global criteria are defined as properties of the whole cluster solution: $\Psi(\mathcal{C})$. Each possible move of a document between two clusters is thus evaluated in terms of its effect on Ψ, which depending on its nature is either minimised or maximised. As shown below, some of the local decision criteria can be expressed by global equivalents and vice versa.

Following Zhao and Karypis (2001) we discuss three types of global criterion functions: internal, external and hybrid criteria.[11]

Internal Criterion Functions. Clustering criteria measuring the *compactness* (cohesiveness) of the clusters are known as *internal criterion functions*. They are by far the most common and calculated as a sum of individual values $E(C)$ for each cluster:

$$\Psi(\mathcal{C}) = \sum_{C_i \in \mathcal{C}} E(C_i), \tag{2.24}$$

In an attempt to bring systematic order into these internal cluster criterion functions $E(C)$, we suggest to write them as a combination of three abstract factors:

$$E(C_i) = w_i \, \frac{S_i}{a_i}, \tag{2.25}$$

where the individual components have the following meaning:

- *Weight* w_i indicates whether clusters are weighted by their size ($w_i = n_i$) or whether all clusters are weighted uniformly ($w = 1$). In most cases the former approach is chosen.
- *Distortion measure* S measures how much a cluster is distorted, i. e. how much it differs from the "perfect" case where all documents in the cluster are identical.

  There are two important choices to be made: (1) the selection of a similarity measure (typically cosine or squared Euclidean distance), and (2) whether to compute pairwise similarities among all cluster members ($\sum \sum s(\mathbf{d}_i, \mathbf{d}_j)$) or whether to compute similarities between the members and a centroid vector ($\sum s(\mathbf{d}, \mathbf{r}^c)$).

[11] We also follow their notation to describe the individual criteria as $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{H}_1$ and $\mathcal{H}_2$.

- *Averaging factor* a depends on S and is used to make the distortion measures comparable among clusters of different sizes. If S is a sum of pairwise similarities ($\sum\sum s(\mathbf{d}_i, \mathbf{d}_j)$), then $a = 2 \cdot n_i^2$ so that S/a indicates the average similarity between two documents. If centroid comparison is chosen, then $a = n_i$ and S/a the average similarity of a document with its cluster centroid.

Various combinations of w, S, a are possible and several can be shown to have equivalent effect. For a comparative examination of the various functions and values for w, S, a see Appendix B.3.

In practice only a few have been used regularly:

- **Classical k-Means.** The classical k-means algorithm (MacQueen, 1967) uses the local decision criterion which assigns each object to the nearest cluster centre as measured by the Euclidean distance. There is an equivalent global decision criterion which is characterised by $w = |C_i|$, $S = \sum \hat{s}_{\text{Euclid}}(\mathbf{d}, \mathbf{r}_i^c)^2$ and $a = n_i$:

$$\begin{aligned} \mathcal{I}_3: \quad \min_{\mathcal{C}} \quad \Psi(\mathcal{C}) &= \sum_{C_i \in \mathcal{C}} E(C_i), \quad \text{with} \\ E(C_i) &= n_i \cdot \frac{\sum_{\mathbf{d}_j \in C_i} \hat{s}_{\text{Euclid}}(\mathbf{d}_j, \mathbf{r}_i^c)^2}{n_i} \\ &= \sum_{\mathbf{d}_j \in C_i} \|\mathbf{d}_j - \mathbf{r}_i^c\|_2^2 . \end{aligned} \tag{2.26}$$

  This is the same as the minimum variances approach known from Equation 2.22. As further shown by Duda and Hart (1973, 219–220), squared Euclidean *centroid* comparison $\sum \hat{s}_{\text{Euclid}}(\mathbf{d}_j, \mathbf{r}_i^c)^2$ (as in Eq. 2.26) is identical to squared Euclidean *pairwise similarity* $\sum\sum \hat{s}_{\text{Euclid}}(\mathbf{d}_j, \mathbf{d}_k)^2$.[12]

  In document clustering, classical k-means has only been moderately successful. Its relative failure is usually attributed to the inadequacy of the Euclidean distance measure in the high-dimensional document space (e. g. Strehl *et al.*, 2000).

- **Vector Space k-Means.** The so-called vector space variant of k-means uses the cosine similarity measure for finding the nearest cluster centroid. It enjoys great popularity for document clustering (Salton and McGill, 1983; Cutting *et al.*, 1992; Dhillon and Modha, 2001; Zhao and Karypis, 2003). The equivalent global cluster criterion is characterised by $w = |C_i|$, $S = s_{\text{Cosine}}(\mathbf{d}, \mathbf{r}^c)$ and $a = |n_i|$:

$$\begin{aligned} \mathcal{I}_2: \quad \max_{\mathcal{C}} \quad \Psi(\mathcal{C}) &= \sum_{C_i \in \mathcal{C}} E(C_i), \quad \text{with} \\ E(C_i) &= n_i \cdot \frac{\sum_{\mathbf{d}_j \in C_i} s_{\text{Cosine}}(\mathbf{d}_j, \mathbf{r}_i^c)}{n_i} \\ &= \sum_{\mathbf{d}_j \in C_i} \cos(\mathbf{d}_j, \mathbf{r}_i^c). \end{aligned} \tag{2.27}$$

[12] The latter is $\mathcal{I}_1$ in the notation of Zhao and Karypis (2001). Note that the centroid used here is the one defined in Equation 2.13 which has *not* unit length even if the document vectors have. Except for an unimportant linear transformation, the two Euclidean measures can also be shown to be equivalent to pairwise cosine similarity *if* the documents are normalised to unit length. See Appendix B.3 for the details of these two equivalences.

An extension of the vector space variant is *toric k-means* for vectors consisting of two or more totally independent parts (Modha and Spangler, 2000).[13] A further generalisation of these and other k-means clustering criteria is described by Modha and Spangler (2003). Frigui and Nasraoui (2004) introduce an interesting algorithm which allows each cluster to have its own individual weighting component for each dimension. Cluster memberships and cluster feature weightings are then simultaneously optimised.

Comparison of Classical and Vector Space Variants. In order to compare the above two variants of k-means, once more following Zhao and Karypis (2001), we define the *composite vector* $\mathbf{y}_i$ of cluster C_i as

$$\mathbf{y}_i = \sum_{\mathbf{d}_j \in C_i} \mathbf{d}_j. \tag{2.28}$$

The objective functions of the two variants of k-means can then be re-stated as follows (see Appendix B.3, cases C and E):

$$\text{Classical } k\text{-means (Euclid):} \quad \Psi(C_1, \ldots, C_k) = n - \sum_{i=1}^{k} \frac{\|\mathbf{y}_i\|_2^2}{n_i}, \tag{2.29}$$

$$\text{Vector space variant (cosine):} \quad \Psi(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{\|\mathbf{y}_i\|_2^2}{\|\mathbf{y}_i\|_2} \qquad \left(= \sum_{i=1}^{k} \|\mathbf{y}_i\|_2\right) \tag{2.30}$$

Abstracting from an irrelevant linear transformation, these two measures differ only in the divisor of $\|\mathbf{y}_i\|_2^2$. In the Euclidean case the divisor is n_i, while for the cosine variant it is $\|\mathbf{y}_i\|_2$, resulting in two different clustering behaviours.

In a geometrical interpretation, the characteristics of the two measures can be visualised by the boundary drawn between two neighbouring clusters in a two-dimensional space (see also Strehl *et al.*, 2000). Of course, the cluster boundary is equal to the set of points which have equal distance (similarity) to both cluster centres (see Figure 2.4).

The points of Euclidean equidistance lie on the dashed perpendicular bisector of the straight line connecting $\mathbf{r}_1^c$ and $\mathbf{r}_2^c$. The points of cosine equidistance rest on the dotted bisector of the angle between $\mathbf{r}_1^c$ and $\mathbf{r}_2^c$. The two boundaries are obviously quite distinct from each other. Only if the centroid vectors have identical length (as happens when they are normalised to unit length) do the two boundaries coincide.

External Criterion Functions. In addition to the traditional internal clustering criteria measuring cluster cohesion, Zhao and Karypis (2001) also suggest a criterion to measure the *separation* between clusters. Their proposed formula results in clusters whose centroids are as far away from the overall collection centroid as possible, with each cluster being weighted by the number of its members. In analogy to Equation 2.28 let there be a composite vector $\mathbf{y}_\mathcal{S}$ of the entire collection:

$$\mathbf{y}_\mathcal{S} = \sum_{i=1}^{k} \mathbf{y}_i = \sum_{j=1}^{n} \mathbf{d}_j \,. \tag{2.31}$$

[13] In Modha and Spangler's example the documents are represented by a combination of three independent vectors: a word space vector, a vector of incoming links and a vector of outgoing links.

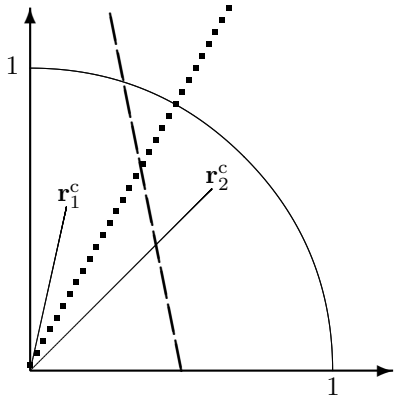


Figure 2.4: **Cluster boundaries.** Boundary between two cluster centres $\mathbf{r}_1^c$ and $\mathbf{r}_2^c$. Dashed line: boundary under Euclidean similarity. Dotted line: boundary under cosine similarity.

The collection centroid $\mathbf{r}_{\mathcal{S}}^c$ can then be defined as follows:

$$\mathbf{r}_{\mathcal{S}}^c = \frac{\mathbf{y}_{\mathcal{S}}}{n}\,. \tag{2.32}$$

A simple external criterion is defined by

$$\begin{aligned} \mathcal{E}_1: \quad \min_{\mathcal{C}} \quad \Psi(\mathcal{C}) &= \sum_{C_i \in \mathcal{C}} E(C_i)\,, \qquad \text{with} \\ E(C_i) &= n_i \cdot \cos(\mathbf{c}_i, \mathbf{c}_{\mathcal{S}}) \\ &= \frac{1}{\|\mathbf{y}_{\mathcal{S}}\|} \cdot n_i \cdot \frac{{\mathbf{y}_i}^T \mathbf{y}_{\mathcal{S}}}{\|\mathbf{y}_i\|}\,. \end{aligned} \tag{2.33}$$

Zhao and Karypis also examine the analogous measure with the Euclidean distance ($\mathcal{E}_2$). It can be shown that this leads to the same result as maximising Euclidean *within-cluster* compactness (see Appendix B.4).

Hybrid Criterion Functions. Hybrid criteria are formulated as combinations of internal and external criteria. Zhao and Karypis (2001) examine two such cases.

The first is a combination of average pairwise cosine similarity[14] and the external criterion defined in Equation 2.33 above, leading to the $\mathcal{H}_1 = \frac{\mathcal{I}_1}{\mathcal{E}_1}$ criterion:

$$\mathcal{H}_1: \quad \max_{\mathcal{C}} \quad \Psi(\mathcal{C}) = \frac{\sum_{i=1}^k \|\mathbf{y}_i\|^2 / n_i}{\sum_{i=1}^k n_i \left({\mathbf{y}_i}^T \mathbf{y}_{\mathcal{S}}\right) / \|\mathbf{y}_i\|}\,. \tag{2.34}$$

Their second hybrid criterion $\mathcal{H}_2 = \frac{\mathcal{I}_2}{\mathcal{E}_1}$ is a combination of average centroid similarity with the cosine (Eq. 2.27) and the same external criterion defined above (Eq. 2.33):

[14] See case A in Appendix B.3. For clarity's purpose it has here been multiplied by the (immaterial) factor two. Remember that average pairwise cosine is related to the average *Euclidean* centroid distance.

$$\mathcal{H}_2: \quad \max_{\mathcal{C}} \quad \Psi(\mathcal{C}) = \frac{\sum \|\mathbf{y}_i\|}{\sum |C_i| \cdot (\mathbf{y}_i{}^T \mathbf{y}_{\mathcal{S}}) / \|\mathbf{y}_i\|} . \tag{2.35}$$

Apart from these internal, external and hybrid functions, the study by Zhao and Karypis (2001) was supplemented with two criterion functions based on the graph model. Their extensive tests showed that internal cosine centroid similarity (Eq. 2.27) and the corresponding hybrid criterion $\mathcal{H}_2$ (Eq. 2.35) performed best, followed by the hybrid criterion $\mathcal{H}_1$ (Eq. 2.34).

For a discussion of further criteria see Everitt (1993).

2.3.1.2 The Initial Partition

Iterative optimisation methods often depend heavily on the initial configuration. The more local extrema there are, the more important becomes the choice of a promising starting point. In other words, the suitable choice of the *initial partition* or the *initial seeds* for the iterative process may be crucial.

In his seminal paper on the k-means algorithm, MacQueen (1967) chose k random *objects* as initial cluster seeds. Related work by Jancey (1966) used random *points* in the feature space, while Forgy (1965) used random *partitions* (cf. also Lance and Williams, 1967b).

Various methods have since been suggested to find better initial partitions (cf. Everitt, 1993). In particular the following methods have been adopted for document clustering:

- *Buckshot.* The *buckshot* algorithm (Cutting *et al.*, 1992) finds initial seeds by clustering a sub-sample of size $\sqrt{kn}$ with the help of a slower but "better" algorithm. They suggest to use group-average agglomerative clustering (see Section 2.4.1) for this purpose. The centroids of the k clusters thus found are then used as seeds for k-means.

  A similar approach is used by Bellot and El-Bèze (2000) who obtain the initial partition through an adaption of single-link clustering.

- *Fractionation.* The *fractionation* algorithm (Cutting *et al.*, 1992) also uses an agglomerative sub-routine to find initial seeds, but it is more complicated since it works with *all* documents and agglomerates them in a multi-step hierarchical process. The algorithm starts by sorting all documents according to a simple criteria (lexical sort on the j^{th} most frequent term in each document) and sequentially dividing them up into $\frac{n}{m}$ "buckets" of fixed size $m > k$. Within these buckets an agglomerative sub-routine is used to obtain pm clusters, with $p \in (0, 1)$ being a fixed "reduction factor". In the next iteration step the documents in these clusters are represented by their centroid. These centroids are again divided into buckets of size pm and these buckets are again clustered. This process is repeated until only the desired number of clusters k remains. This final partition is then used to generate the initial seeds for k-means. Cutting *et al.* show that this algorithm has rectangular running time $O(mn)$ and is thus less complex than ordinary hierarchical methods.

- *Iterative Refinement.* An iterative refinement method is suggested by Bradley and Fayyad (1998). They take several small sub-samples which they cluster with k-means. From the resulting cluster centroids they compute in a second step the refined initial centroids, which are then used for the whole collection.

- *Random Sampling.* This widely used method does not actually result in an initial clustering, but aims to overcome the same sensitivity problem. Instead of a single k-means run, multiple runs are performed with different random seeds. In the end only the result of the best such run is kept (e. g. Zhao and Karypis, 2003). Of course, there is still no guarantee that a good initial starting point has been found.

Larsen and Aone (1999) compare random partition, buckshot and fractionation, finding comparatively little difference in the quality of the results as long as the clusters were updated *continuously* (see the following section).

2.3.1.3 Allocation of Documents to Clusters

After the establishment of an initial configuration, the iterative process of (re)allocating the individual objects sets in. Here too a number of small but sometimes important distinctions can be made. One important decision is whether to re-calculate the cluster centroids continuously (i. e. immediately after the re-assignment of a single object) or non-continuously (i. e. only after one round of assigning each object to the nearest cluster has been completed). Non-continuous updating used to be the standard choice for quite a while, but recent studies have clearly preferred a random-order continuous updating strategy (MacQueen, 1967; Larsen and Aone, 1999; Steinbach *et al.*, 2000; Zhao and Karypis, 2003). In the latter case we must distinguish between "moving" individual documents from one cluster to the next and "clearing" the clusters between iterations, meaning that each iteration starts with empty clusters ("bare centroids") before each object is freshly assigned to the nearest centroid. Larsen and Aone moreover suggest a method called *vector average damping* which lends a higher weight to the documents which have just joined a cluster *vis-à-vis* those that have been assigned earlier. The theoretical foundation for doing so is unclear, but the procedure appears to have been successful.

Several authors (Larsen and Aone, 1999; Bellot and El-Bèze, 2000; Wang and Kitsuregawa, 2001, and others) use a minimum similarity threshold which must be met before an object can be assigned to a cluster. Outliers not meeting the criterion are either dumped altogether or collected in a special "junk" cluster (Hearst and Pedersen, 1996).

2.3.1.4 Stop Criterion and Post-Processing

The choice of a suitable stop criterion depends on the individual purpose and specifics of the algorithm. Since k-means and related approaches can be shown to converge in finite time, the simplest stop criterion is to wait until no further changes occur. An alternative is to define a minimum threshold for the cluster criterion function—either an absolute value or a minimal improvement required per iteration. Time-critical applications often use a fixed value of iterations (e. g. Cutting *et al.*, 1992). Larsen and Aone (1999) find that with continuous updating already a single iteration usually leads to reasonably good results.

Cutting *et al.* (1992) and Larsen and Aone (1999) also describe further ways to *refine* the cluster solution obtained by k-means. Their algorithms identify potentially "bad" clusters which are *split* and "close" clusters which are *joined*. For a more detailed approach to cluster merging and splitting see Ding and He (2002). Liu *et al.* (2002) describe another refinement method which identifies *discriminative* features for each cluster.[15] Based on these discriminative features the documents are (re)assigned to individual clusters. Just like the original k-means algorithm this

[15] They call a feature discriminative if its *discriminative feature metric (DFM)* exceeds a certain minimum threshold θ:

$$\mathrm{DFM}(f_j) = \log \frac{\max_i \; g_j(C_i)}{\left[\sum_i g_j(C_i) - \max_i \; g_j(C_i)\right] / (k-1)} > \theta, \tag{2.36}$$

process is iterated until it converges, with the discriminative features freshly determined after each iteration.

2.3.2 Alternative Clustering Algorithms

Apart from the iterative partitional methods just examined and the yet to be treated group of hierarchical algorithms (Section 2.4), countless further clustering algorithms exist. This section aims to introduce a few which share a certain significance for document clustering.

2.3.2.1 Single-Pass Sequential Algorithms

Under heavy time and space constraints the use of a sequential *single-pass* algorithm may become mandatory. A variety of variants exists (cf. Kaufman and Rousseeuw, 1990, 156–157), but the basic principle remains the same: the data set is traversed just once; each object is added to the "nearest" cluster unless the distance exceeds a certain minimum threshold, in which case a new cluster is built from this document. The approach is efficient but has a number of drawbacks: the result is strongly dependent on the order of the data, it is difficult to decide upon a suitable threshold, it is impossible to predict the number of clusters that result and no theoretical foundation exists in the form of a global optimisation criterion.

A similar method has been suggested among others by MacQueen (1967) as a variant of the original k-means algorithm. His algorithm includes tests whether two clusters exceed a certain similarity threshold, in which case the clusters are merged.

Such *incremental* clustering algorithms may be especially useful in a Web search context where the documents arrive one by one and need to be clustered quickly (Hatzivassiloglou *et al.*, 2000; Hammouda, 2001). See also BIRCH (Section 2.4.1.5) and DBSCAN (Section 2.3.2.2). Unfortunately, the cluster solutions thus produced are often of limited quality only.

A specific and fast algorithm for document clustering is *Suffix Tree Clustering* (Zamir and Etzioni, 1998, 1999). Scanning the documents one by one, a data structure called suffix tree is constantly updated. It serves to identify phrases common to different documents. In a second processing phase clusters are distilled from the phrases and corresponding documents. The algorithm appears to be a promising method for fast on-line clustering.

2.3.2.2 Density-Based Clustering

Density-based clustering relies on a spatial notion of "density". A data point p lies within a dense area if there is a minimal number m of points which lie in an ε-neighbourhood of p. Separate conditions apply for the points at the border of a dense region. Densely-connected points form a cluster. The standard *DBSCAN*[16] algorithm (Ester *et al.*, 1996) has a complexity of the order $O(n \log n)$. Its great advantage (compared to the common partitional clustering methods) is that is not restricted to *convex* clusters but can recognise arbitrarily shaped clusters (cf. Figure 2.5). On the other hand it is difficult to identify appropriate parameters m and ε. DBSCAN is rarely used for document clustering, one exception being the work by Wen *et al.* (2001) and Wen and Zhang (2003).

---

where $g_j(C_i)$ indicates the frequency of feature f_j in cluster C_i. It is not entirely clear whether these frequencies should be document frequencies or actual term frequencies, nor if absolute or relative frequencies are meant (which can make a difference if the clusters differ in size).

[16] *DBSCAN* is an acronym for *Density Based Spatial Clustering of Applications with Noise.*

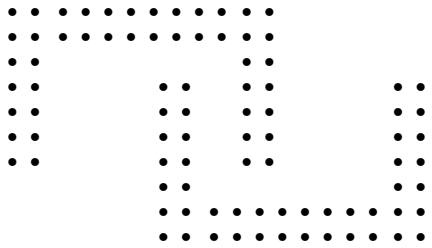


Figure 2.5: **Two non-convex clusters.** They cannot be identified by traditional k-means and related partitional methods. For density-based approaches they pose no problem.

2.3.2.3 Probabilistic Models

The k-means, the hierarchical and many other algorithms share a lack of a firm statistical foundation, a deficit that is specifically addressed by the class of probabilistic cluster algorithms.

They view the clustering problem as one of *identifying hidden component functions which generate the observed data*. Once identified, each component function can then be interpreted as forming a cluster. For a detailed introduction to these *finite mixture* models see Titterington *et al.* (1985). Below follows just a short summary (see also Duda and Hart, 1973; Everitt, 1993; Bradley *et al.*, 1998).

The goal is to find function parameters $\Theta = (\theta_1, \ldots, \theta_k)$ and weights $\Pi = (\pi_1, \ldots, \pi_k)$ which lead to an optimal estimation of the following *probability density function*:

$$p(\mathbf{d} \,|\, \Theta, \Pi) = \pi_1 f_1(\mathbf{d} \,|\, \theta_1) + \ldots + \pi_k f_k(\mathbf{d} \,|\, \theta_k),\ \mathbf{d} \in \mathcal{S}. \tag{2.37}$$

The functions $f_1 \ldots f_k$ denote the individual component densities, while $\pi_1 \ldots \pi_k$ denote different weights or *prior probabilities (mixing parameters)* of the components (with $\sum_{i=1}^{k} \pi_i = 1;\ \pi_i \geq 0$).

Since no direct solutions exist for estimating Θ and Π, iterative optimisation methods must be used. The most popular of these is the *expectation-maximisation (EM) method* (Dempster *et al.*, 1977), a maximum-likelihood approach which maximises the following log-likelihood function:

$$\mathcal{L}(\mathcal{S} \,|\, \Theta, \Pi) = \sum_{\mathbf{d} \in \mathcal{S}} \log \left[\sum_{i=1}^{k} \pi_i f_i(\mathbf{d} \,|\, \theta_i) \right]. \tag{2.38}$$

The algorithm iteratively and alternately re-estimates the mixture probabilities π_i in an *expectation* and the model parameters θ_i in a *maximisation* step until $\mathcal{L}$ converges to a local maximum. In the end cluster membership probabilities can be calculated for each document according to Bayes' principle:

$$\begin{aligned} p(\mathbf{d} \in C_i) \quad = \quad p(C_i \,|\, \mathbf{d}) \quad &= \quad \frac{p(C_i)\, p(\mathbf{d} \,|\, C_i)}{p(\mathbf{d})} \\ &= \quad \frac{\pi_i\, p(\mathbf{d} \,|\, f_i)}{\sum_{j=1..k} \pi_j\, p(\mathbf{d} \,|\, f_j)}. \end{aligned} \tag{2.39}$$

Choosing for each document the cluster with the highest probability results in an non-overlapping partition of all documents.

A powerful algorithm for solving this very general model is *AutoClass* (Cheeseman *et al.*, 1988; Cheeseman and Stutz, 1996). In document clustering it has only occasionally been used (e. g. Goldszmidt and Sahami, 1998; Boley *et al.*, 1999a), restricted and extended models (see below) having enjoyed more popularity.

In practice, a number of simplifications are frequently encountered:

- The component functions $f_1 \dots f_k$ are almost always assumed to be multivariate normal distributions (Gaussian distributions), which reduces the parameter space to mean vectors and covariance matrices (see e. g. Liu *et al.*, 2002):

$$\theta_i = (\mu_i, \Sigma_i). \tag{2.40}$$

  The individual component density functions thus have this common form:

$$f(\mathbf{d} \,|\, \mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^m |\Sigma_i|}} \, e^{-\frac{1}{2}(\mathbf{d}-\mu_i)^T (\Sigma_i)^{-1} (\mathbf{d}-\mu_i)}, \tag{2.41}$$

  with $|\Sigma_i|$ being the determinant of the covariance matrix, Σ^{-1} the matrix inverse, π the number Pi and T indicating vector transposition.[17]

- During EM-optimisation, instead of viewing the documents as the sum of probabilities over all clusters, they are assigned to one single cluster (e. g. Fasulo, 1999).

- Often the assumption is made that the individual attributes (terms) occur *independently* of each other within a document. This assumption is also inherent to the classical bag-of-words model used in various other algorithms.

- The number of components (clusters) is fixed at k. This can often be problematical when k does not correspond to the number of underlying components. Smyth (1996) discusses various optimisation methods aimed at obtaining the best value for k (see also Liu *et al.*, 2002).

It can be shown that by adding two further restrictions the EM-approach can be reduced to the classical k-means algorithm (Bradley and Fayyad, 1998). They are:

[17] The EM-steps for multivariate Gaussians then look as follows (Liu *et al.*, 2002):

Expectation step. The new expectations $\hat{\pi}$ are calculated from the old probabilities π and model estimates:

$$\hat{\pi}_i = \frac{1}{n} \sum_{j=1}^{n} p(C_i|\mathbf{d}_j)\,, \quad \text{with} \tag{2.42}$$

$$p(C_i|\mathbf{d}_j) = \frac{\pi_i f(\mathbf{d}_j|\mu_i, \Sigma_i)}{\sum_{l=1}^{k} \pi_l f(\mathbf{d}_j|\mu_l, \Sigma_l)}\,. \tag{2.43}$$

Maximisation step. The model parameters are updated to maximise the log-likelihood:

$$\hat{\mu}_i = \frac{\sum_{j=1}^{n} p(C_i|\mathbf{d}_j)\, \mathbf{d}_j}{\sum_{j=1}^{n} p(C_i|\mathbf{d}_j)}\,, \tag{2.44}$$

$$\hat{\Sigma}_i = \frac{\sum_{j=1}^{n} p(C_i|\mathbf{d}_j)(\mathbf{d}_j - \mu_i)(\mathbf{d}_j - \mu_i)^T}{\sum_{j=1}^{n} p(C_i|\mathbf{d}_j)}\,. \tag{2.45}$$

- $\Sigma = \sigma^2 I$ where I is the identity matrix; i. e. the clusters are modelled as *spherical* Gaussian distributions,

- the mixture weights are all equal ($\pi_1, ..., \pi_k = 1/k$).

Fasulo (1999) reviews work by Banfield and Raftery (1993) which explores various other restrictions on the general mixture model. Bradley *et al.* (1998) present a method to scale EM-clustering to large databases. A *latent class* approach (cf. *Latent Semantic Analysis*, Section 3.4.3.2) is discussed in a probabilistic setting by Hofmann (1999a,b) and Vinokourov and Girolami (2000).

Despite their theoretical superiority, mixture models are not universally preferred over k-means or hierarchical models. One reason is the often large number of free parameters, which not only increases computation cost but also leads to numerous undesirable local maxima, without there being a sure way to overcome them. Furthermore, the EM-algorithm sometimes converges only very slowly and different initial seeds may be required (Everitt, 1993). Finally, mixture models fail if the underlying components are not *identifiable* (Duda and Hart, 1973, 190–191, 205).

2.3.2.4 Self-Organising Maps (Kohonen Maps)

Neural networks in the form of self-organising feature maps (Kohonen, 1984, 2001) are a popular feature reduction technique and enjoy a large following in unsupervised learning research. For experiments in the document clustering area see Kaski *et al.* (1998); Pullwitt and Der (2001); Bakus *et al.* (2002); Henderson *et al.* (2002a,b).[18]

WEBSOM (Kaski *et al.*, 1998) is a two-level approach to document clustering with SOMs. The first level maps the words onto a reduced "word category map" (in a semantic sense) by taking the immediate neighbouring words into account. The second level maps the documents as encoded by these word categories onto a final two-dimensional document map, where similar documents are represented by the same or by nearby nodes.

SOM algorithms impose an order onto the clusters by arranging them in a two-dimensional field, with related clusters being arranged together. This approach makes only sense if a sufficiently large number of clusters is sought.

2.3.2.5 Genetic Algorithms

Another class of methods for multi-dimensional optimisation processes which have gained considerable fame in artificial intelligence are the *genetic algorithms (GAs)*. See Cristofor and Simovici (2001) for a recent overview of GAs for clustering. Experiments with documents are rare and Jones *et al.* (1995) conclude that they are less suitable for this area than are other algorithms.

2.3.2.6 Decision Trees

Bellot and El-Bèze (2000) describe a method to grow an *unsupervised decision tree* for clustering. This tree does not cluster documents but the individual sentences. There are familiarities with *suffix tree clustering* mentioned above (page 33), but the idea has been little explored so far.

[18] Another neural network technique used for document clustering is Adaptive Resonance Theory under Constraints (*ART-C*) (He *et al.*, 2002, 2003).

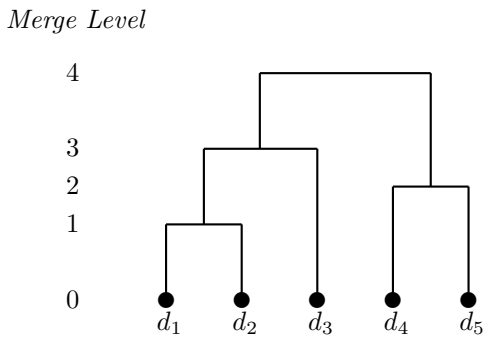


Figure 2.6: **A dendrogram**, visualising a hierarchical arrangement of objects in a binary tree. The further up a horizontal connection, the larger the distance between the two nodes. The vertical axis can either be a pure ordinal scale or it can be used to indicate those distances.

2.4 Hierarchical Algorithms

The underlying principle of hierarchical clustering algorithms is simple and elegant: given a criterion function (measure of similarity) all objects can be arranged deterministically in a *(binary) tree* which provides a complete order for all objects. The tree has the individual objects as its leaves, while the nodes can be interpreted as clusters containing the objects found when the tree is further traversed downwards. The tree is typically depicted as a *dendrogram* (see Figure 2.6). The higher the level at which two nodes are connected, the larger the distance between them.

Hierarchical clustering strategies work either in a bottom-up (*agglomerative*) direction (Section 2.4.1) or in a top-down (*divisive*) direction (Section 2.4.2). Two other crucial factors (recurring in the discussion below) are the criterion function and the stop criterion (respectively the manner of transforming the tree into an actual set of clusters).

2.4.1 Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering (HAC) is one of the best explored cluster algorithms of all, with a particularly long tradition in taxonomy (Sneath and Sokal, 1973). Many important facets of its application in document clustering were treated in a seminal paper by Willett (1988).

In HAC all n objects are initially placed in n separate clusters. These clusters are then iteratively merged two at a time until one single cluster remains or some other stop criterion is met. The "merging history" can be depicted in a dendrogram (see Figure 2.6).

From an information-theoretic point of view, HAC is an iterative information reduction technique which is stopped when the desired granularity is reached (Ward, 1963).

2.4.1.1 Cluster Criterion Function

The key aspect of HAC is the choice of a *criterion function* which governs at each step the choice of clusters to be merged. As with the iterative partitional methods (Section 2.3.1), there exist both local decision criteria for merging the two "most similar" clusters and globally defined optimisation functions (cf. Willett, 1988; Everitt, 1993; Zhao and Karypis, 2002, 2003).

2.4.1.2 Local Criterion Functions

Single-Linkage. Single-linkage clustering is based on the nearest-neighbour principle (Eq. 2.16). It can be implemented by efficient algorithms (for an overview, see Willett, 1988); however, the quality of the resulting clustering has often been found to be unsatisfactory because of the "chaining effect" (forming loose clusters with little internal cohesion; Willett, 1988). A number of methods have been developed to dampen the chaining effect, one recent suggestion being *CURE* which was briefly described in Section 2.2.2.

Complete-Linkage. Complete-linkage clustering is based on the furthest-neighbour principle (Eq. 2.18). It is computationally more expensive than single-linkage but works better in the presence of noise. It tends to build more compact clusters.

Group-Average (UPGMA[19]). Group-average hierarchical clustering is based on the group average similarity measure (Eq. 2.20). It has been adopted successfully in numerous domains.

Weighted Group-Average (WPGMA[20]). WPGMA is the complement of UPGMA, but rarely seen, especially in the document domain. In WPGMA the individual objects in a cluster do not contribute equally to the average similarity but are weighted by the step number at which they joined the cluster. At each merging step the two groups are given equal weight in the newly merged cluster, meaning that the objects in the smaller group have a proportionally higher weight in future average calculations.

The difference is illustrated by these two equations:

$$\begin{aligned} &\underline{\text{UPGMA :}} \\ S(C_1, \{C_2 \cup C_3\}) &= \sum_{\mathbf{d}_a \in C_1} \frac{1}{n_1(n_2 n_3)} \left[\sum_{\mathbf{d}_b \in C_2} s(\mathbf{d}_a, \mathbf{d}_b) + \sum_{\mathbf{d}_b \in C_3} s(\mathbf{d}_a, \mathbf{d}_b) \right]; \quad (2.46) \\ &\underline{\text{WPGMA :}} \\ S(C_1, \{C_2 \cup C_3\}) &= \frac{1}{2} S(C_1, C_2) + \frac{1}{2} S(C_1, C_3) \\ &= \sum_{\mathbf{d}_a \in C_1} \frac{1}{2n_1} \left[\sum_{\mathbf{d}_b \in C_2} \frac{s(\mathbf{d}_a, \mathbf{d}_b)}{n_2} + \sum_{\mathbf{d}_b \in C_3} \frac{s(\mathbf{d}_a, \mathbf{d}_b)}{n_3} \right]. \quad (2.47) \end{aligned}$$

Centroid Comparison (UPGMC[21]). Hierarchical clustering based on centroids measures similarity between two clusters by comparing the two centroids (cf. Eq. 2.13, resp. Eq. 2.14). Jain and Dubes (1988, 80) recommend that it be only used in conjunction with Euclidean distance.

Weighted Centroid Comparison (WPGMC[22]). In strict analogy with the group average method, there exists an even rarer weighted centroids method.

[19] UPGMA: Unweighted Pairwise Group Method with Averages.
[20] WPGMA: Weighted Pairwise Group Method with Averages.
[21] UPGMC: Unweighted Pairwise Group Method with Centroids.
[22] WPGMC: Weighted Pairwise Group Method with Centroids.

Method	α_1	α_2	β	γ
Single-Linkage	0.5	0.5	0	-0.5
Complete-Linkage	0.5	0.5	0	0.5
UPGMA	$\frac{n_1}{n_1+n_2}$	$\frac{n_2}{n_1+n_2}$	0	0
WPGMA	0.5	0.5	0	0
UPGMC	$\frac{n_1}{n_1+n_2}$	$\frac{n_2}{n_1+n_2}$	$\frac{-n_1 n_2}{(n_1+n_2)^2}$	0
WPGMC	0.5	0.5	-0.25	0
Ward's Method	$\frac{n_1+n_3}{n_1+n_2+n_3}$	$\frac{n_2+n_3}{n_1+n_2+n_3}$	$\frac{-n_3}{n_1+n_2+n_3}$	0

Table 2.1: **The Lance-Williams coefficients** for local decision criteria (see Eq. 2.48).

Ward's Method. The popular clustering method suggested by Ward (1963) derives from the minimum variance criterion introduced in Equation 2.21. At each step the algorithm chooses to merge those two clusters resulting in a minimal overall increase of "information loss".

This method has performed well in various different applications, but one of the disadvantages of Ward's method is that it tends to produce spherical clusters even when other shapes would be more appropriate. For an application of Ward's method to document clustering see El-Hamdouchi and Willett (1986).

The local criteria just discussed can be summarised in the cluster similarity update scheme introduced by Lance and Williams (1967a):

$$\begin{aligned} S(C_1 \cup C_2, C_3) &= \alpha_1 \cdot S_{1,3} + \alpha_2 \cdot S_{2,3} + \beta \cdot S_{1,2} + \gamma \cdot |S_{1,3} - S_{2,3}|\,, \\ \text{with} \quad S_{i,j} &= S(C_i, C_j)\,. \end{aligned} \tag{2.48}$$

The individual methods can then be written in terms of α_1, α_2, β and γ as in Table 2.1.

2.4.1.3 Global Criterion Functions

The global criterion functions $\mathcal{I}_1 \ldots \mathcal{H}_2$ discussed in detail in Section 2.3.1 can also be used for hierarchical clustering. In particular, it should be noted that Ward's method is equivalent to the $\mathcal{I}_3$ resp. $\mathcal{I}_1$ criterion.

Zhao and Karypis (2002) compared these global and several local criterion functions in the context of different document clustering algorithms. For HAC they obtained the best results with UPGMA, followed closely by $\mathcal{I}_2$. Tombros *et al.* (2002) tested the four most popular local criteria (single-link, complete-link, UPGMA and Ward's method) in a dynamic retrieval environment. They also found that group average (UPGMA) performed best, thus confirming the results of some older static retrieval studies.

Other global criteria include the CHAMELEON algorithm (Karypis *et al.*, 1999) which is a combination of both inter- and intra-cluster connectivity measures (working on graphs) and the *information bottleneck method* (Slonim and Tishby, 2000, cf. Section 3.4.3.1) which merges at

each step those two clusters resulting in a minimal loss of *mutual information* between objects and features.

2.4.1.4 Stop Criterion

The HAC algorithm is usually terminated when... (or the complete cluster tree "cut" at the point where...)

- ...the number of clusters has been reduced to a predefined external value k (*fixed number of clusters*), or
- ...the distance between the next two clusters to be merged exceeds a certain predefined threshold (*criterion-driven number of clusters*).

Both cases require a user-defined external parameter to be set and it is also possible to combine the two. The problem of fixing k is further discussed in Section 2.5.3. If the cluster tree is very skewed ("outlier" documents or tiny clusters being merged in only at the very top of the tree), it may be desirable (depending on the application) to prune the tree first in order to achieve a more balanced cluster solution.

2.4.1.5 Modifications to HAC

Time complexity is a constant issue in HAC and much effort has been invested into the design of time- and storage-efficient HAC implementations. For most criteria algorithms of the order $O(n^2 \log n)$ exist, for some the complexity can be reduced to $O(n^2)$.

In earlier years single-linkage has been particularly popular because of the availability of efficient quadratic algorithms (cf. Willett, 1988; Maarek *et al.*, 2000). Maarek *et al.* propose a simplified algorithm for complete-linkage by which complexity can also be reduced to quadratic order. They replace the binary dendrogram by a more flexible, non-binary tree structure.

BIRCH[23] is another simplified method for large data samples which uses a non-binary balanced tree structure. Rather than merging individual clusters, BIRCH scans the data linearly and updates the tree structure after every element (Zhang *et al.*, 1996).

Constrained Agglomerative Clustering is a new two-phase method suggested by Zhao and Karypis (2002, 2003) in the context of document clustering. By combining partitional and agglomerative clustering, they aim to exploit the advantages offered by the two approaches. In the first phase they make use of a partitional clustering algorithm to divide the data set into a number of large partitions which are individually clustered by an agglomerative algorithm. After the partitions have been clustered internally, the partitions themselves are also clustered agglomeratively.

2.4.2 Hierarchical Divisive Clustering

Hierarchical agglomerative clustering is perhaps the best-explored clustering method of all, but comparatively little attention has been paid so far to the divisive "top-down" alternative. Recent document clustering studies (Zhao and Karypis, 2002, 2003) indicate, however, that we may see more of that in the future.

Hierarchical Divisive Clustering (HDC) starts with one big cluster containing all the objects, and proceeds by iteratively splitting clusters until the desired granularity is reached. One of the problems of the divisive approach is that by accident two very similar documents may easily be

[23] *BIRCH* is an acronym for *Balanced Iterative Reducing and Clustering using Hierarchies.*

split into two different clusters at an early stage, after which there is usually no easy remedy to correct the mistake (Everitt, 1993, 55, also 82–88).

Below follow brief descriptions of two recent HDC approaches used for document clustering. Basically, these methods must answer two questions: (1) which cluster to split next, and (2) how to split that cluster. In addition, a stop criterion may be defined in analogy to HAC.

Principal Direction Divisive Partitioning (PDDP) (Moore *et al.*, 1997; Boley *et al.*, 1999a,b). In PDDP the next cluster to be split is the one with the highest *scatter value* (i. e. the *error sum of squares*—see Equation 2.21).

Cluster splitting is performed by translating the documents around the origin of the vector space and then computing the *principal direction* (direction of maximum variance) of the object-feature matrix of the cluster.[24] The hyperplane orthogonal to the principal direction and going through the origin is used to split the documents into two groups/clusters.

The stop criterion is reached when the scatter value of all individual clusters is less than the scatter value of all the cluster centroids.

Repeated Bi-Sections (Steinbach *et al.*, 2000). The bisecting approach makes use of an ordinary, non-hierarchical partitional clustering method to split clusters. Steinbach *et al.* choose at each step the largest cluster, which is split into two groups by the popular k-means algorithm with $k = 2$ (Section 2.3.1). Since the result of k-means depends on the (random) initialisation, at each step several tentative bi-sections are tested and only the solution resulting in the clusters with the highest average pairwise intra-cluster similarity (measured by the cosine similarity) is actually used.[25] As a stop criterion Steinbach *et al.* chose a fixed number of clusters.

The approach of Steinbach *et al.* was further refined and generalised by Zhao and Karypis (2001, 2002, 2003) who tested a variety of global cluster criteria and different methods for determining the next cluster to be split. In particular, instead of simply splitting the largest cluster (which tends to produce clusters of similar size), they recommend as an alternative to take all clusters into consideration and then choose the split resulting in the best value for the criterion function. Since their algorithms and software were used for our own experiments, further details can be found in Section 4.2.

The repeated bi-sections approach just described combines various elements of hierarchical and non-hierarchical algorithms. Combined with an effective post-processing (refinement) method, it appears to be an excellent answer to the document clustering task (Zhao and Karypis, 2003).

2.5 Cluster Solutions and Properties

The present section is devoted to a number of issues primarily related to the properties of the final cluster solution. In particular, a number of methods will be mentioned that deviate from the simple "one document, one cluster" principle.

[24] The principal direction is defined as the eigenvector corresponding to the largest eigenvalue of the covariance matrix. The method has thus a certain similarity with *Latent Semantic Analysis* (Section 3.4.3.2).

[25] Average pairwise intra-cluster similarity measured by the cosine is equal to the square of the Euclidean length of the non-normalised cluster centroid (derived from normalised object vectors). See Appendix B.3, case A.

2.5.1 Shape

As has been mentioned in the previous discussion, some of the standard clustering algorithms fail to recognise clusters with unusual, non-convex forms or densities (cf. Figure 2.5). It has therefore been argued that k-means is a poor choice as it is limited to the creation of convex spherical clusters (*Voronoi/Dirichlet partitions*).

It is not immediately clear whether this restriction is relevant for document clustering which is characterised by very high-dimensional spaces. Next to nothing is known of the spatial distribution of real-world text documents in the feature spaces typically used. Intuition and previous experiments would suggest, however, that groups occurring in real-world document sets do not have such extraordinary shapes that could not be recognised by the traditional algorithms.

2.5.2 Structure

Significant static characteristics of a cluster solution can be described by these four attributes:

Completeness. In most of the traditional algorithms and in our discussion so far it has been assumed that cluster solutions are *complete*, i. e. that *every* object is assigned to a cluster:

$$\bigcup C_1 \dots C_k \equiv \mathcal{S}\,. \tag{2.49}$$

However, so-called outliers may have undesirable effects on the cluster solution. In particular, if the aim is not to accommodate all patterns but rather to extract the main groups of a data set (e. g. the main topics of a corpus or the "nuggets"; Ertöz *et al.*, 2003), it might be better to discard outliers or collect them in a separate "junk cluster" as is done by Hearst and Pedersen (1996) and Maarek *et al.* (2000). For a discussion of outliers and their timely elimination see also Guha *et al.* (2003).

A different approach is applied by Wang and Kitsuregawa (2001). In a pre-processing step they try to filter out "low-quality" documents (measured in their case by in- and outgoing links). The actual cluster solution is thus restricted to "high-quality" documents, the assumption being that they better describe the interesting characteristics of the collection.

Exclusivity. Traditional cluster solutions are *exclusive*, i. e. each object is assigned to exactly one cluster, resulting in a partition without overlap:

$$C_i \cap C_j \equiv \emptyset\,, \quad \text{with} \quad i, j \in \{1, \dots, k\} \wedge i \neq j\,. \tag{2.50}$$

However, in a rich textual environment this can be a severe restriction since a single document may deal with several different topics and may therefore justifiably belong to more than one cluster with equal right. The restriction cannot be overcome in traditional hierarchical methods, but for approaches based on an intermediate structure—such as sentences or phrases (Zamir and Etzioni, 1998; Bellot and El-Bèze, 2000)—assigning a document to multiple clusters is no problem. Probabilistic models (Section 2.3.2.3) can also be used to assign each object to all those clusters meeting a minimal probability threshold.

Degree of Membership. A concept closely related to exclusivity is the *degree of membership* of an object in a cluster. So far we have assumed this to be a binary decision, but this need not be the case. Often we will find situations where certain objects are "better" or "more typical" members of a cluster than others. It is thus possible to define a degree of membership function δ on a continuous scale rather than the customary binary scale:

$$\delta(D_i, C_j) \in [0, 1]. \tag{2.51}$$

The concept of non-binary degrees of membership can come itself in two different variations:

- $\sum_{j=1}^{k} \delta(D_i, C_j) = 1$: The membership values for each object sum to 1, which is typical of the probabilistic models discussed in Section 2.3.2.3.
- $\sum_{j=1}^{k} \delta(D_i, C_j) \in [0, k]$: The membership values for all object-cluster pairs are independent of each other and may sum to arbitrary values for each document. This principle is used by clustering methods derived from *fuzzy logic* (Zadeh, 1965). Unlike the probabilistic model, *fuzzy clustering* permits each object to be highly relevant to several clusters. See Miyamoto (1990) for an overview of fuzzy methods in information retrieval and clustering. A fast fuzzy version of the k-medoid algorithm is demonstrated by Krishnapuram *et al.* (1999), a method based on *data bubbles* is discussed by Newton and O'Brien (2002).

Of course, a continuous membership function δ is only useful if it can be communicated to the end-user in a meaningful way (e. g. by visualisation) or if it can otherwise be used in post-processing. In particular within an IR system, the danger of confusing the user by multiple occurrences of the same document must be addressed.

Nestedness. As already seen, cluster solutions can be either nested (hierarchical) and unnested (flat). It has been argued (e. g. Maarek *et al.*, 2000) that for browsing a document collection, a hierarchical solution is both more informative and more effective than a flat one since in a hierarchy traversal of the tree takes logarithmic time as opposed to linear time for a flat partition. On the other hand, creating a hierarchy is usually more expensive.

For evaluation purposes, nested cluster solutions are usually transformed into a flat partition as the latter are by far better explored.

2.5.3 Model Selection Problem

One of the most crucial questions in many real-world cluster applications (including document clustering) is determining a suitable number of clusters k, also known as the *model selection problem* (or at least a crucial part thereof). Without *a priori* knowledge there is no simple way of knowing that number.

The following methods exist for arriving at a useful value for k:

External choice. Often k is simply fixed by the application designer. His choice is usually based on either experience or the restrictions of the application-interface. Alternatively, it may be left to the end-user to choose an appropriate value of k.

From a practical perspective, a fixed value k is often the simplest and fastest solution.

Search for k. More sophisticated approaches involve testing several values for k and choose the one performing best on a global validation scale. Various procedures have been suggested to achieve this.

Pelleg and Moore (2000) use the *Bayesian Information Criterion (BIC)* to choose between different models (cluster solutions) M_k:

$$BIC(M_k) = \mathcal{L}_k(\mathcal{S}) - \frac{p_k}{2} \cdot \log n, \tag{2.52}$$

where $\mathcal{L}_k$ is the log-likelihood of the data $\mathcal{S}$ at the maximum-likelihood point, while p_k is the number of parameters in M_k (for k-means this is $k - 1$ class probabilities + $m \cdot k$

centroid coordinates + $m \cdot k$ individual variances). The model M_k maximising the BIC is eventually chosen.

Liu *et al.* (2002) suggest to perform multiple randomly initiated runs for each value of k under consideration and then select that value of k whose cluster solution is the *most stable* (i. e. the value for which the solutions from different runs are most similar to each other and which is thus believed to be most likely to correspond to the "true" solution). Similarity between solutions for a given k is measured by a mutual information metric.

Duda and Hart (1973, 241–243) describe an approach based on hypothesis testing. Smyth (1996) suggests a Monte-Carlo cross-validation technique. Li *et al.* (2004) present an approach based on the eigenvalues of the HH^T matrix. Rezaee *et al.* (1998) give an overview of several methods used in fuzzy clustering.

A number of validation techniques discussed in Section 2.6.4 can also be used on a similar basis to tackle the model selection problem.

Criterion-driven determination of k. In hierarchical algorithms the number of clusters can often be determined by a certain *stop criterion* such as a maximum distance beyond which no two clusters can be merged. The number of clusters is therefore implicitly defined via the criterion function and a corresponding external threshold. The latter, however, remains subjected to the arbitrary choice of the designer or end-user.

Similar criterion-driven values of k may also result from partitional algorithms where thresholds can be set which govern splitting and merging of clusters during the main process (e. g. with single-pass clustering or in a general post-processing refinement phase). In these cases k is outside the control of the user, but the threshold must again be determined externally.

Although much effort has gone into determining the "right" number of clusters, in the document clustering domain it may actually be an ill-posed question. As shown in a practical user study (Macskassy *et al.*, 1998), when asked to cluster documents manually, test persons showed widely different preferences. The study further showed that although different users preferred different numbers of clusters, they were usually quite consistent in the number of clusters they chose in different situations. This may indicate that k as a user-defined parameter may after all be a sensible solution in IR. Further studies in that direction would be helpful.

2.6 Evaluation and Validation

This section discusses different methods and formulae commonly used to measure and compare the quality of different cluster solutions. Following Frakes and Baeza-Yates (1992) we distinguish between *validation* as a means of comparing clusters and cluster solutions and *evaluation* as a means of comparing different algorithms or, in our case, different document representation techniques. The two are very closely related and evaluation is often simply relying on particular validation techniques. Another important aspect of evaluation is algorithmic complexity (Section 2.6.7).

2.6.1 Approaches to Evaluation and Validation

The problem of clustering validity is extensively treated in Jain and Dubes (1988, Chapter 4). They differentiate between three types of objects that can be considered: hierarchical groupings, flat partitions and single clusters. Furthermore, they differentiate between three types of criteria, to which two more IR-specific categories have been appended in the list below:

- *External Criteria.* If data-independent labels (categories) are available, the question may be asked of how well a given cluster solution corresponds to these external labels. In document clustering manually labelled collections are a very popular means of evaluating new algorithms, but care is required because of human errors in the labelling as well as the possibility of the algorithm discovering a valid structure that had been overlooked by the human indexer. Again, to be able to make reliable statements a suitable baseline must be defined. External criteria have also been used repeatedly to compare the relative merits of two or more cluster solutions.

- *Internal Criteria.* Here the principal question is: how good does a particular cluster solution or individual cluster fit the data? Various measures or indices exist, but to obtain a meaningful answer they must be put into relation with a carefully chosen *null hypothesis*, for which appropriate *baseline distributions* must be derived. In practice these baseline distributions are rarely available and must thus be estimated with suitable statistical models (e. g. Monte Carlo simulations). Once such a baseline is determined, it is possible to assess how well a specific clustering solution differs from random cluster assignments.

- *Relative Criteria.* Relative criteria are used to answer the relative merits of two (or more) cluster solutions in the absence of external data (labels). This includes questions such as which is the "true" number of clusters k underlying a certain data set.

- *Ranked-List Comparison.* If clustering is specifically used as a means of organising documents in a large collection as a preparation for cluster-based retrieval—as was mostly the case in the earlier years of the document clustering discipline—the cluster solution can be evaluated by traditional IR instruments such as *precision* and *recall*.

- *End-User Criteria.* Regardless of all the theoretical background, the best evaluation method for practical applications (such as search engine results clustering) is the user's judgement. After all, the usefulness of any clustering system depends on the users' satisfaction and it can never be fully measured by abstract coefficients. However, reliably measuring users' satisfaction is extremely difficult in itself (Stefanowski and Weiss, 2003) and so far no suitable methodology seems to have been suggested. However, for a recent study comparing human and algorithmic perception of mere similarity (without clustering) see Lee *et al.* (2005).

The subsequent sections cover some of the widely used measures and indices, without aiming at a complete overview. For a thorough treatment compare Jain and Dubes (1988) and Halkidi *et al.* (2002).

2.6.2 External Criteria

Usually, pre-classified document collections are flat and thus most practical validity indices apply to *partitions*. See Jain and Dubes (1988) for how external criteria may work with *hierarchies* and Rezaee *et al.* (1998) and Halkidi *et al.* (2001) for *fuzzy clustering*. For partitions a detailed treatment can also be found in Jain and Dubes (1988) and Dom (2002). Our discussion focusses on a few central aspects.

Let there be l *a priori* labelled classes. Each object belongs to one such class L_i:

$$L_1 \dots L_l \subset \mathcal{S} \tag{2.53}$$

$$\wedge \quad \forall\, i, j : i \neq j \rightarrow L_i \cap L_j = \emptyset \tag{2.54}$$

$$\wedge \quad \bigcup L_i = \mathcal{S}\,,\ i \in \{1 \dots l\}\,. \tag{2.55}$$

The subject under examination being a flat exclusive partition, each object also belongs to exactly one cluster C_j. This membership information can be summarised in a two-dimensional *contingency table* $\mathcal{H} \equiv \{h(L,C)\}$, where $h(L_i, C_j)$ is the number of documents in cluster C_j with label L_i.

2.6.2.1 Overlap Indices

From the contingency table $\mathcal{H}$ the following numbers can be defined:[26]

$$a_{00} = \sum_{i,j} \binom{h(L_i, C_j)}{2}, \tag{2.56}$$

$$a_{01} = \sum_{i} \binom{|L_i|}{2} - a_{00}, \tag{2.57}$$

$$a_{10} = \sum_{i} \binom{|C_i|}{2} - a_{00}, \tag{2.58}$$

$$a_{11} = \binom{n}{2} - (a_{00} + a_{01} + a_{10}). \tag{2.59}$$

Then a_{00} is the number of object *pairs* where both objects belong to the same label *and* to the same cluster. a_{01} is the number of pairs that belong to the same label but different clusters, while a_{10} are the pairs in the same cluster but with different labels. Finally, a_{11} is the number of object pairs that share neither label nor cluster. In IR parlance a_{00} are the *true positives*, a_{10} the *false positives*, a_{01} the *false negatives* and a_{11} the *true negatives*.

Further we define:

$$\tilde{a} = (a_{00} + a_{01})(a_{00} + a_{10}). \tag{2.60}$$

Based on these coefficients, four common external validity indices are known (Jain and Dubes, 1988; Dom, 2002):

Rand: $$\frac{a_{00} + a_{11}}{\binom{n}{2}}\,. \tag{2.61}$$

Jaccard: $$\frac{a_{00}}{a_{00} + a_{01} + a_{10}}\,. \tag{2.62}$$

Fowlkes/Mallows: $$\frac{a_{00}}{\sqrt{\tilde{a}}}\,. \tag{2.63}$$

Γ statistic: $$\frac{a_{00}\binom{n}{2} - \tilde{a}}{\sqrt{\tilde{a}\left(\binom{n}{2} - a_{00} - a_{01}\right)\left(\binom{n}{2} - a_{00} - a_{10}\right)}}\,. \tag{2.64}$$

All these indices increase with rising overlap between labels and clusters. See Jain and Dubes (1988) for how to construct a baseline to decide whether or not an index value indicates a *significant* overlap.

2.6.2.2 The $\mathcal{Q}_0$-Measure

In recent years indices based on the *mutual information* concept have increasingly gained attention (e. g. Strehl *et al.*, 2000). Dom (2002) worked out a general *cluster quality measure* $\mathcal{Q}_0$, which

[26] $\binom{x}{y} = \frac{x!}{y!(x-y)!}$.

is applicable also to cases where the number of clusters and classes need not be equal. $\mathcal{Q}_0$ is the sum of two components: the first measures the *empirical conditional entropy* $\hat{H}(\mathcal{L}|\mathcal{C})$ between the labels $\mathcal{L}$ and the clusters $\mathcal{C}$ (which is equivalent to the empirical mutual information), the second component measures the cost of encoding the contingency table $\mathcal{H}$ and favours solutions with a smaller number of clusters. The smaller the value of $\mathcal{Q}_0$, the better is the corresponding cluster solution.

$$\mathcal{Q}_0(\mathcal{L},\mathcal{C}) = \hat{H}(\mathcal{L}|\mathcal{C}) + \frac{1}{n}\sum_{i=1}^{k} \log \binom{|C_i| + |\mathcal{L}| - 1}{|\mathcal{L}| - 1}, \tag{2.65}$$

$$\text{with } \hat{H}(\mathcal{L}|\mathcal{C}) = -\sum_{i=1}^{l}\sum_{j=1}^{k} \frac{h(L_i,C_j)}{n} \log \frac{h(L_i,C_j)}{|C_j|}. \tag{2.66}$$

2.6.2.3 Distance Measures

A number of theoretically less well-founded measures have also been adopted, including a series of measures based on precision and recall (see the following sub-section) and some other distance measures:

- *Average Distance:* Bradley and Fayyad (1998) use the average distance between the class (label) centroids and the cluster centroids as a measure of how well a cluster solution matches the "ground truth".
- *Editing Distance:* Pantel and Lin (2002) propose to measure the difference between labels and clusters by the number of operations that are needed to transform the latter into the former. The three possible editing operations are (a) merging two clusters, (b) moving an object from one cluster to another, and (c) copying an object from one cluster to another. Unlike the indices discussed above, editing distance also works with overlapping clusters. Pantel and Lin suggest to transform this distance measure $dist(\mathcal{C},\mathcal{L})$ into a cluster quality measure as follows:
$$Q(\mathcal{C},\mathcal{L}) = 1 - \frac{dist(\mathcal{C},\mathcal{L})}{dist(\mathcal{B},\mathcal{L})}, \tag{2.67}$$
where $\mathcal{B}$ is a baseline clustering with each document in its own cluster.

2.6.2.4 Precision and Recall Measures

Several indices have been suggested based on precision and recall. For a detailed discussion see Yang (1999).

For individual clusters and labels, we define precision P and recall R as follows:

$$\text{Precision:} \quad P(L_i,C_j) = \frac{h(L_i,C_j)}{|C_j|}, \tag{2.68}$$

$$\text{Recall:} \quad R(L_i,C_j) = \frac{h(L_i,C_j)}{|L_j|}. \tag{2.69}$$

The following measures are regularly seen in practice:

Purity. The purity measure used e. g. by Strehl *et al.* (2000) and Zhao and Karypis (2001) is the maximum possible precision for each cluster:

$$\text{Purity:} \quad \tilde{P}(C_i) = \max_{j=1\ldots l} P(L_j,C_i). \tag{2.70}$$

When these purity values are summed up over all clusters, they can be weighted by cluster size or not.

Zhao and Karypis (2001) use weighted purity values:

$$\text{Weighted Purity:} \quad \Psi(\mathcal{C}|\mathcal{L}) = \sum_{i=1}^{k} \frac{n_i}{n} \tilde{P}(C_i)\,. \tag{2.71}$$

Modha and Spangler (2003), following Yang (1999), call this *micro-precision/micro-recall*. This measure is equivalent to "accuracy" and also to the "classification error" (e. g. Goldszmidt and Sahami, 1998) which counts the number of objects in a cluster that have a "minority label" ($\sum_i n_i - \max_j h(L_j, C_i)$).

The unweighted alternative is called *macro-precision* by Yang (1999) and Modha and Spangler (2003):

$$\text{Unweighted Purity:} \quad \Psi(\mathcal{C}|\mathcal{L}) = \frac{1}{k} \sum_{i=1}^{k} \tilde{P}(C_i)\,. \tag{2.72}$$

They further define a *macro-recall* as follows:

$$\text{Macro-Recall:} \quad \Psi(\mathcal{C}|\mathcal{L}) = \frac{1}{k} \sum_{i=1}^{k} R(\bar{L}_i, C_i)\,, \tag{2.73}$$

$$\text{with} \quad \bar{L}_i = \arg\max_{L_j} \tilde{P}(L_j, C_i)\,. \tag{2.74}$$

Entropy. One of the most popular individual measures is the entropy measure. Like a number of other measures it needs to be handled carefully in situations with a variable number of clusters k because a solution with each document in its own cluster will always score optimally. In terms of precision, cluster entropy can be defined as follows:

$$\text{Entropy:} \quad E(C_i) = -\sum_{j=1}^{l} P(L_j, C_i) \log P(L_j, C_i)\,. \tag{2.75}$$

Optionally, the entropy values can be standardised to an $[0, 1]$ scale by multiplying them with the constant factor $\frac{1}{\log l}$ (e. g. Zhao and Karypis, 2001).

An overall weighted entropy measure is then defined as

$$\text{Weighted Entropy:} \quad \Psi(\mathcal{C}|\mathcal{L}) = \sum_{i=1}^{k} \frac{n_i}{n} E(C_i)\,. \tag{2.76}$$

F-Measure. Another highly popular measure is the F-score (Larsen and Aone, 1999) which is based on the F-measure in information retrieval $F = \frac{(\beta^2+1)PR}{\beta^2 P+R}$ (Jardine and van Rijsbergen, 1971). Larsen and Aone set $\beta = 1$, a value also used in various subsequent clustering studies (Steinbach *et al.*, 2000; Bakus *et al.*, 2002; Zhao and Karypis, 2003). The F-score of a cluster is its maximum possible F-measure:

$$\textit{F-Score}(C_i) = \max_{j \in \{1 \ldots l\}} \frac{2\, P(L_j, C_i)\, R(L_j, C_i)}{P(L_j, C_i) + R(L_j, C_i)}. \tag{2.77}$$

Again, an overall evaluation function can be built by taking weighted or unweighted averages of individual F-scores.

To sum up, a large number of methods exist to assess a cluster solution in the presence of an "objective" ground truth. Often they are used to evaluate performance of algorithms or feature representations in experimental setups. In actual applications, however, such *a priori* knowledge usually does not exist as it would make clustering superfluous.[27]

2.6.3 Internal Criteria

In the absence of external labels, cluster solutions can still be assessed by internal criterion functions such as cluster compactness and cluster separation (and generally all those global criteria discussed in Sections 2.3.1 and 2.4.1). However, establishing suitable baselines (null hypotheses) for comparison is very difficult in these cases. Therefore two other internal criteria are more popular, one for partitions and one for hierarchies:

Hubert's Γ Statistic. Given two $n \times n$ proximity tables $\mathcal{X} = [X(i,j)]$ and $\mathcal{Y} = [Y(i,j)]$ Hubert's Γ is defined as follows:

$$\Gamma(\mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X(i,j)\, Y(i,j). \tag{2.78}$$

In normalised form, Γ is the sample correlation coefficient of the entries in the two matrices. If we take $\mathcal{X}$ to be a cosine or Euclidean proximity matrix of the objects in $\mathcal{S}$, and $\mathcal{Y}$ to be a matrix indicating whether two objects are in the same cluster (proximity 0) or in two different clusters (proximity 1), then useful values for Γ can be calculated as well as an appropriate random baseline.

Cophenetic Correlation Coefficient (CPCC). The *cophenetic correlation coefficient* also uses an ordinary geometric proximity table $\mathcal{X}$, while the cophenetic proximity table $\mathcal{Y}$ measures cluster proximity between two documents by the dendrogram level at which they appear in the same cluster for the first time. If the number of entries above the main diagonal of the matrices is defined as $M = n(n-1)/2$ and the respective means as $m_X = (1/M) \sum_{i,j \in n} X(i,j)$ and $m_Y = (1/M) \sum_{i,j \in n} Y(i,j)$, then

$$CPCC = \frac{(1/M) \sum_{i,j \in n} X(i,j), Y(i,j) - (m_X m_Y)}{\sqrt{(1/M) \sum_{i,j \in n} X(i,j)^2 - m_X} \cdot \sqrt{(1/M) \sum_{i,j \in n} Y(i,j)^2 - m_Y}}\,. \tag{2.79}$$

CPCC assumes values between −1 and 1. The nearer to 1, the better the clustering.

For details about the generation of the baseline values which allow to distinguish between random results and significant clusterings see Jain and Dubes (1988).

2.6.4 Relative Criteria

Assessing the relative merits of two cluster solutions does not require any baselines, removing the main restriction of the previous section. Typically, cluster solutions from different parameter settings are compared, and the goal is to determine the "best" of these solutions. Individual parameters (such as k, the number of clusters) can be adjusted through multiple tests and the observation of the behaviour of the index criterion with varying parameter values. Depending on the nature of the index that is used, either a global maximum/minimum occurs or at least

[27] For evaluations with *two* external "ground truths" (two classification schemes) see Rosell *et al.* (2004).

a significant "knee" in the criterion function, which points to the optimal parameter value. Numerous criteria have been tested in practice (Jain and Dubes, 1988; Halkidi *et al.*, 2002), of which we discuss two representative examples.[28]

Davies–Bouldin Index. The index suggested by Davies and Bouldin (1979) for different values of k is typically initiated as follows to capture the concepts of cluster separation (denominator) and cluster compactness (nominator):

$$\mathrm{DB}(\mathcal{C}_k) = \frac{1}{k}\sum_{i=1}^{k} \max_{j\neq i\in\{1\ldots k} \frac{s_i + s_j}{\|\mathbf{r}_i^c - \mathbf{r}_j^c\|_2}, \tag{2.80}$$

with s_i the square root of the average error (within-cluster variance) of cluster C_i (cf. Eq. 2.22):

$$\begin{aligned} s_i &= \sqrt{V(C_i)} \\ &= \sqrt{\frac{\sum_{\mathbf{d}\in C_i}(\mathbf{d}-\mathbf{r}_i^c)^T(\mathbf{d}-\mathbf{r}_i^c)}{n_i}}\,. \end{aligned} \tag{2.81}$$

By maximising the $(s_i+s_j)/(\|\mathbf{r}_i-\mathbf{r}_j\|)$ term, the worst constellation for each cluster in the cluster solution is considered. The smaller the Davies–Bouldin index, the better therefore a given cluster solution.

SD Index. A recently suggested relative criterion is the *SD index* (Halkidi *et al.*, 2000). It also relies on measures of compactness and separation, but is calculated differently:[29]

- *Average Scattering.* Cluster compactness is measured by the average "scattering" of the clusters which is defined as the length of the *variance vector* σ of each cluster (operator $\otimes$ denotes component-wise vector multiplication between vectors of equal length, i. e. $\mathbf{a}\otimes\mathbf{b} \equiv I\mathbf{a}^T\mathbf{b}$, with I the identity matrix):

$$\mathrm{Scatter}(\mathcal{C}_k) = \frac{1}{k}\cdot\sum_{i=1}^{k}\frac{\|\sigma(C_i)\|_2}{\|\sigma(\mathcal{S})\|_2}, \tag{2.82}$$

$$\text{with}\quad \sigma(C_i) = \frac{1}{n_i}\sum_{d\in C_i}(\mathbf{d}-\mathbf{r}_i^c)\otimes(\mathbf{d}-\mathbf{r}_i^c)\,. \tag{2.83}$$

The chosen formula gives stronger weight to deviations in individual positions as shown by the following comparison of the variable parts of the compactness formulae:

$$\mathrm{BD}: \quad s_i = \frac{\sqrt{n_i}}{n_i}\sqrt{\sum_{d_j\in C_i}\sum_{f\in\{1\ldots m\}}(d_{jf}-r_f^c)^2}\,, \tag{2.84}$$

$$\mathrm{SD}: \quad \|\sigma(C_i)\|_2 = \frac{1}{n_i}\sqrt{\sum_{d_j\in C_i}\sum_{f\in\{1\ldots m\}}(d_{jf}-r_f^c)^4}\,. \tag{2.85}$$

[28] See also the criteria mentioned in the preceding section.

[29] For a further disquisition on compactness and separation see He *et al.* (2003).

- *Total Separation.* The total separation between clusters is measured as in the following formula. Again the individual clusters are given more weight by summing the inverses of their total distances from each other rather than just inverting the entire sum:

$$\begin{aligned} \mathrm{Dis}(\mathcal{C}_k) &= \frac{D_{max}}{D_{min}} \sum_{i=1}^{k} \left[\sum_{j=1}^{k} \|\mathbf{c}_i - \mathbf{c}_j\|_2 \right]^{-1}, \qquad (2.86) \\ \text{with} \quad D_{max} &= \max_{i \neq j \in 1 \ldots k} \|\mathbf{c}_i - \mathbf{c}_j\|_2 , \\ D_{min} &= \min_{i \neq j \in 1 \ldots k} \|\mathbf{c}_i - \mathbf{c}_j\|_2 . \end{aligned}$$

Given these two factors, the SD index is calculated as a linear combination

$$\mathrm{SD}(\mathcal{C}_k) = \alpha \cdot \mathrm{Scatter}(\mathcal{C}_k) + \mathrm{Dis}(\mathcal{C}_k). \qquad (2.87)$$

Halkidi *et al.* suggest to set the weighting factor α to $\mathrm{Dis}(\mathcal{C}_{\mathrm{k_{max}}})$, where k_{max} is the maximum number of clusters under consideration. The smaller the two components of the SD index are, the better is the clustering. Whether or not the SD index can establish itself as a viable alternative to the older relative criteria remains to be seen.

As both the DB and SD indices are independent of the cluster number k, their respective values for different k allow the determination of an "optimal" number of clusters k.

2.6.5 Ranked-List Criteria

If clustering is used to produce a ranked list (e. g. in a search engine) and if corresponding external relevance judgements are available, a number of further indices can be calculated.

Tombros *et al.* (2002), following Jardine and van Rijsbergen (1971), compute the E-values of all clusters and choose the value of the most effective cluster (i. e. the one with the least E-value) as clustering validity index *MK1*. The E-measure is defined as $E = 1-F$, with F being computed from precision and recall as in Equation 2.77. The optimal cluster efficiency *MK1* is compared with the ranked-list measures *MK1-k* and *MK3*, which are defined as follows: If k is the number of documents in the most effective cluster, *MK1-k* is the E-value of the top k documents on the ranked-list. *MK3*, on the other hand, is the optimal value attainable by the ranked-list (i. e. if it is cut off at the point where E reaches a minimum). Depending on the values used for β in Equation 2.77, they obtain quite different verdicts for cluster-based versus ranked-list retrieval.

Zamir and Etzioni (1998) compare clustering and ranked-list by taking precision values of the top 10% of returned documents. For clustering, the clusters are first sorted by recall and then documents are added first from top to bottom of the best cluster, then from the second cluster, etc. until 10% of the whole set are reached. See also Hearst and Pedersen (1996) and Bellot and El-Bèze (2000) for related efficiency measures.

Most of these methods depend on the assumption that the user is able to immediately recognise and pick the most relevant clusters. A short survey by Hearst and Pedersen (1996) showed that this was the case in about 80% of all cases. Of course, ranked-list comparisons are based on a one-sided view of document clustering, concentrating on the relevance/irrelevance aspect, but ignoring other purposes such as document or concept organisation, exploratory search, etc.

2.6.6 End-User Criteria

The number of actual user studies on clustering usefulness is very small. Both Hearst and Pedersen (1996) and Zamir and Etzioni (1999) make use of log analysis, but as yet no reliable methodology seems to have been developed.

2.6.7 Complexity and Performance

The complexity of various clustering algorithms ranges from $O(n)$ for most partitional to $O(n^2)$, $O(n^2 \log n)$ or even $O(n^3)$ for the hierarchical algorithms. For a comparison of the complexity as well as other characteristics of a large number of clustering algorithms see Steinbach *et al.* (2000); Strehl *et al.* (2000); Halkidi *et al.* (2001); Zhao and Karypis (2002).

Comparing actual running times is notoriously difficult and strongly dependent on the hardware used. For on-line systems fast response times are obviously highly desirable. Working with document snippets, Zamir and Etzioni (1998) showed that using linear algorithms it was possible to cluster up to 600 documents in less than six seconds on a Pentium 200 processor. Of course, in view of the constant progresses in hardware development, such a figure has only anecdotal significance. Questions of scalability in a document context are addressed explicitly by Boley *et al.* (1999b); Weiss *et al.* (2000a); Dhillon and Modha (2001); Dhillon *et al.* (2001).

Chapter 3

Document Representation

Number is the first principle
and the matter in things
and in their conditions and states;
and the odd and the even are elements of number,
and of these the one is infinite
and the other finite,
and unity is the product of both of them,
for it is both odd and even,
and number arises from unity,
and the whole heaven, as has been said, is numbers.

Pythagoras (in Aristotle's *Metaphysics*)

At least for computers the Pythagorean world view holds: all is numbers. The translation of an ordinary text document into numbers, more precisely into a vector, is therefore necessary and it forms the topic of this chapter. Success or failure of a document clustering application is often highly dependent on the choice of a suitable representation method.

The following discussion of representation techniques is dominated by the ultimate purpose, document clustering. However, there is a very strong overlap between the methods used here and those applied in related fields such as text categorisation or text storage and retrieval. All are concerned with finding a "good" feature space in which to accurately represent the documents, their main characteristics and their similarities/dissimilarities.

We begin by an exposition of the initial document space (Section 3.1), followed by an examination of functions transforming a linear document into a feature vector ("vectorisation", Section 3.2). These vectors are usually weighted and/or further refined (Sections 3.3 and 3.4). We then conclude the discussion with the implications of these methods for clustering (Section 3.5) and a section on the visualisation of document clusters (Section 3.6).

3.1 Document Space

Before examining different methods of projecting documents into feature spaces, we shall look at the original document space and some of its properties.

3.1.1 Restricted vs. Open Topics

Much document clustering work has been performed and evaluated on dedicated collections, with documents stemming from a unique or from very similar sources (e. g. Usenet postings, Reuters news bulletins) or belonging to a single general domain (e. g. medical abstracts). In literature, little use has been made of specific knowledge about such restricted domains. An exception is the domain-dependent ontology investigated by Hotho *et al.* (2002).

For cluster applications on top of a Web search engine, such methods would hardly be appropriate since Web indices typically know no topical boundaries. With texts being retrieved from a wealth of different sources and domains, the amount of formal and semantic ambiguity sharply increases, making a succinct representation more difficult. At the same time, it is especially in these open and highly ambiguous domains that clustering can be of the greatest service by organising very heterogenous search results.

With regard to topics, a vexing problem are those documents that deal with several, perhaps even totally unrelated topics. Such a document is not only difficult to assign to a single cluster, it also distorts the delineation between clusters. One approach to get to grips with this problem is to allow fuzzy or overlapping clusters (see Section 2.5.2).

3.1.2 Collection Size

The size n of the document collection $\mathcal{S}$ is crucial for the speed of every clustering algorithm. In particular *ad-hoc* clustering of search results must take the user's notorious impatience into account. But even for off-line clustering, most of the more sophisticated algorithms become impractical if the documents number millions or more.

For a majority of the approaches presented hereafter the trade-off between quantity (speed) and quality (accuracy) is resolved in favour of the latter. For work aimed explicitly at developing fast algorithms for very large text collections see work by Weiss *et al.* (2000a), Dhillon *et al.* (2001) and Newton and O'Brien (2002).[1] In some instances, an upper limit for the number of documents to be clustered is set. For example, Zamir and Etzioni (1998, 1999) and Tombros *et al.* (2002) restrict themselves to the top-ranked x documents from their retrieval systems.

In almost none of the clustering approaches the number of documents n plays a role other than that of a very basic parameter (e. g. for estimating the number of desired clusters k). However, for *classification* tasks, Perlich *et al.* (2003) have shown that the choice of a suitable algorithm may depend on the collection size. It might thus be interesting to investigate whether there are parallels in the field of *clustering*, i. e. whether the optimal choice of clustering and/or representation method might depend on the observed set size n. At least it makes sense to define a minimum number of documents for clustering to take place at all. If there are just five documents, a simple list is just as useful as the most sophisticated clustering.

3.1.3 Document Size

A property quite typical of the document domain is that the individual objects can have arbitrary and highly varying lengths. Particularly on the World Wide Web, but also in more specialised collections, document sizes differ enormously. Since clustering algorithms usually rely on a measure of similarity between two documents, the question must be addressed if and how to take these length differences into account (cf. Section 3.3).

Because of the limited availability of full-texts and a lack of adequate hardware, the IR systems of the 1970s and 1980s worked mostly with simple document titles or at best with abstracts. With

[1] Of course, fast clustering algorithms for very large data sets were also explored for other domains, leading to algorithms such as BIRCH (Zhang *et al.*, 1996), CURE (Guha *et al.*, 1998) or DBSCAN (Ester *et al.*, 1996).

the digital availability of most resources and the increased storage and processing capabilities of modern days, most present-day retrieval systems use full-text representations. However, for certain time-critical applications a restriction to partial documents must be considered. For instance, Zamir and Etzioni (1998); Krishnapuram *et al.* (1999); Joshi and Jiang (2001) show that the short descriptions ("snippets") returned by search engines can suffice to produce viable results.

3.1.4 Language

Many document representation algorithms such as *stemming* and *stopword removal* (see further below) rely on at least a rudimentary language model. It is therefore desirable to identify the language of the documents to be clustered. Most work has been done for English texts, but papers on clustering in other languages (French, German, Polish) are also available (Bellot and El-Bèze, 2000; Hotho *et al.*, 2002; Stefanowski and Weiss, 2003). It is to be expected that different languages favour different document representation methods.

In the era of the World Wide Web, multi-lingual document collections are a common occurrence. Little effort has gone into clustering such document sets. In the absence of a special effort in that direction it is to be expected that most algorithms would tend to cluster documents of different languages into separate clusters. Silva *et al.* (2001) report on an experiment with clustering of legislation texts in three different languages. The clustering algorithm had no trouble recognising and separating the different languages. An attempt to resolve the cross-lingual difficulties is the highly ambitious *Universal Networking Language* (UNL) project. Should it be made to work, cross-lingual clustering would become a realistic possibility (Choudhary and Bhattacharyya, 2002).

3.1.5 Intra- and Inter-Document Structure

Document *retrieval* algorithms have long been known to exploit specifics of the document structure. In the simplest case this happens when only title and abstract fields are indexed but not the whole body of the text. Other methods lend extra weight to terms in the title or those that are typographically emphasised. In richly-formatted documents such as HTML texts "keywords", "description" and other meta tags offer themselves as candidates for special treatment. In document *clustering* few efforts have so far been undertaken to make use of these internal structures. One example is Weiss *et al.* (2000a), who filter out a few keywords from the text while making sure that all the *title* words are still included in the feature set.

The idea of exploiting internal document structures for clustering ought to be further explored. However, in view of the great diversity of documents on the World Wide Web and the great number of pages where tags and meta tags are used inconsistently or incorrectly (and not seldom abusively—cf. Henzinger *et al.*, 2002, for this and other challenges faced by the major search engines) caution is required.

Inter-document relationships as a means of identifying good representations and good clusters have received more attention. In particular hyperlink analysis has been famously used to obtain measures of document "quality" or "relevance" (Kleinberg, 1998; Page *et al.*, 1998).[2] The number of common direct or indirect hyperlinks is a natural measure of similarity between two documents. It is not surprising that various studies have attempted to make use of these relationships. Pirolli *et al.* (1996), Pitkow and Pirolli (1997), Weiss *et al.* (1996) and Modha and Spangler (2000) present promising approaches to document clustering based on a combination of content and

[2]These approaches are inspired by the ancient concept of co-citation analysis (cf. also van Rijsbergen, 1979, 61–62, Wen and Zhang, 2003).

link information. Wang and Kitsuregawa (2001) and Noel *et al.* (2003) present approaches exclusively relying on link analysis. Further experiments must show whether this is a reliable clustering method; one of its undeniable advantages is its relative language-independence.[3]

3.1.6 Document Quality

Compared to earlier document clustering studies which were based on a relatively narrow and uniform set of documents, modern approaches in a *WWW* environment face not only great varieties in document length and structure but also in document *quality*[4]. Search engine companies have put an enormous effort into establishing good algorithms for separating high-quality from low-quality documents, so that the list of results is topped by documents which are considered both highly relevant and of high quality (cf. Henzinger *et al.*, 2002). Rough measures of quality can be gained by different means. A popular method is the aforementioned link analysis (Kleinberg, 1998; Page *et al.*, 1998, etc.) but interesting experiments have also been conducted with stylistic elements (Karlgren, 1999). Finally, document *type* (e. g. HTML, PDF or PostScript files) could also be used as an indicator of a document's quality, though the usual caveats apply.[5]

So far document quality has hardly ever been considered a factor in document clustering but various applications can be imagined. For example, a clustering algorithm might first filter out the high-quality documents of the result set and calculate a good clustering solution based on just these. Afterwards the documents of lower quality could be filled in, without having a direct influence on the cluster definitions (cf. the "initial partition" task discussed in Section 2.3.1.2).

3.1.7 Knowledge about the Document Universe

Clustering algorithms can be divided into those which work exclusively on the document selection $\mathcal{S}$ and those which make use of additional base information from the entire document universe Ω. Such information can consist of term statistics over the whole text universe or information about the documents in Ω which link into $\mathcal{S}$ or are linked to from $\mathcal{S}$.

The majority of clustering methods investigated at present is inspired by text-based clustering of Web search engine results. In this case the document universe Ω is often intractable, so that usually no distinction between $\mathcal{S}$ and Ω is made.

3.2 Document Vectorisation

This section deals with different methods of mapping a text document (i. e. a linear string of word tokens) onto a feature *vector*.

The general form of a vectorisation function is

$$\mathbf{d_i} = \tau(D_i, \zeta)\,, \tag{3.1}$$

wherein ζ denotes the assumptions and parameters of the language and real-world model (e. g. a list of named-entities, a grammar, an ontology, etc.). It should be noted that the rest of the document collection $\mathcal{S}$ is usually *not* considered part of ζ. Considerations which also involve the other documents are reserved to a separate step, the refinement and weighting phase (Sections 3.3 and 3.4).

[3] In subsequent work Wang and Kitsuregawa also included content information (Wang and Kitsuregawa, 2002).

[4] No attempt at defining *quality* will be made here; a general notion is sufficient for the present purposes.

[5] In certain contexts (such as academic research) document type might even be a useful *primary* clustering criterion.

The following sections describe several different τ functions. In practice, they can also be combined, leading to a document vector consisting of multiple distinct sub-vectors:

$$\mathbf{d_i} = \begin{pmatrix} \tau_1(D_i, \zeta_1) \\ \vdots \\ \tau_z(D_i, \zeta_z) \end{pmatrix}. \tag{3.2}$$

3.2.1 Bag-of-Words

In the simplest and most popular case a document is represented by the unordered set of individual word tokens making up that document (the "bag-of-words"). Identical tokens are grouped together and summed. The total feature space $\mathcal{F}$ thus equals the set of unique words (*word types*) occurring in $\mathcal{S}$. The sequence of features is arbitrary; if a feature is absent from a document, the corresponding value of the document vector is set to zero.

In early IR applications binary vectors were often adopted (van Rijsbergen, 1979). They just indicate whether a term is present or absent in a given text. In recent times, and especially for clustering applications, multi-valued non-negative vectors have been preferred, where each value indicates the number of occurrences of an individual term in the text. Given the large variety of words in natural language it comes as no surprise that these vectors have a very large number of dimensions and that they are usually very sparse (in a typical document vector more than 95% of all values are zeroes).

The bag-of-words model (BOW) is the standard for thousands of IR applications. Splitting a document into word tokens is very fast and the frequencies are easily added up. Moreover, no external model ζ is required.[6]

3.2.2 Annotated Bag-of-Words

The simple BOW model can be refined by annotating the individual tokens with additional, context-dependent information before summing. Such annotations usually refer to the position and function of the individual token within the document and the most typical application is *part-of-speech (POS) tagging*.

By annotating words with their parts-of-speech (word classes, lexical tags) before transforming them into a vector, we can preserve potentially important information about their function in the text which can be further exploited in the refinement phase. Besides, POS tags allow a differentiation between words that look the same but have different functions (e. g. "can" as a verb and "can" as a noun).

Research into automated POS tagging dates back to the 1960s. See Jurafsky and Martin (2000, chapter 8) for an overview of part-of-speech classes, tagging and the three major forms of algorithms: rule-based, stochastic and transformation-based tagging.

For clustering, POS tagging on its own is usually not worth the (considerable) extra effort. However, it is a prerequisite for several of the more sophisticated refinement methods discussed further below.[7]

[6] We will not discuss *tokenisation/lexical analysis*—i. e. delineating and identifying tokens, handling punctuation, etc.—any further, even though no uniquely correct method exists but several related procedures. See Fox (1992), Manning and Schütze (1999, 124–134) and Baeza-Yates and Ribeiro-Neto (1999, 165–167) for further details.

[7] For a discussion of several annotation strategies see also work by Kanejiya *et al.* (2004).

3.2.3 Word Sequences

Various attempts have been made to create features of higher complexity than just word tokens. Many of these methods take the individual word-context into account and thus aim to reduce the quite significant information loss incurred by vectorisation. Such models are often described as "phrase-models". To distinguish the linguistic sense of a phrase (a group of words forming a grammatical unit in the syntax of a sentence) from an arbitrary statical view (a sequence of words), we call the former *syntactic phrases* and the latter *statistical phrases* (Croft *et al.*, 1991).

3.2.3.1 Statistical Phrases

The use of phrases in a non-linguistic sense has been long known in document retrieval, but has often produced rather disappointing results (Salton and McGill, 1983; Fagan, 1989; Croft *et al.*, 1991). Recently some fresh attempts have been made for clustering.

Fixed length (N-grams). Skogmar and Olsson (2002) report on a small-scale experiment using bi- and trigrams instead of unigrams (words), but they find no improvement in the quality of the clustering. Modha and Spangler (2003) combine single words with 2- and 3-word phrases (keeping of all three categories only those features that meet a certain minimal document frequency condition). No comparison with a "single words only" approach is reported.[8] Word-order is usually discarded in n-gram (or n-word combinations) indexing (Chu *et al.*, 2003).

Broder *et al.* (1997) use a shifting N-gram approach (e.g. with $N = 3$ the text "To be or not to be" is translated into the four tuples ⟨To, be, or⟩, ⟨be, or, not⟩, ⟨or, not, to⟩, ⟨not, to, be⟩). For their goal—identifying identical or nearly identical Web documents—this computationally rather expensive approach seemed to work since a much cruder measure of similarity can be applied than for user-oriented document clustering.[9]

Letter-based N-gram models which do not examine complete words or sentences but groups of successive letters (*sliding N-grams*) were used by Larocca Neto *et al.* (2000), apparently with positive results. They do not indicate which values of N they used, but values in the range 3 to 5 seem reasonable. For instance, with $N = 3$ the word "Data" is transformed into the set of features {_DA, DAT, ATA, TA_}.

Joshi and Jiang (2001) also use sliding N-grams on the letter level, but without an extra "end token" (such as "_" above). By using binary weighting and a simple overlap coefficient they claim to overcome minor spelling mistakes as well as the language barrier.

Arbitrary length. Several methods have been suggested for extracting useful word sequences of arbitrary length. Zamir and Etzioni (1998) introduced an efficient algorithm based on a tree structure. Their work on *suffix tree clustering* was taken up by Hammouda and Kamel (2002), Stefanowski and Weiss (2003) and others.

Hannappel *et al.* (1999) use the LZW algorithm for data compression (Ziv and Lempel, 1977; Welch, 1984) to identify frequent phrases which they cluster by a simple mechanism.

Bakus *et al.* (2002) use a "hierarchical phrase grammar" on a training set to identify relevant phrases. These are determined by merging tokens (either individual terms or previously

[8] Some more experience is available from text *classification*. Work by Mladenić and Grobelnik (1998) showed no further improvements by adding N-grams with $N > 3$. Caropreso *et al.* (2001) describe an N-gram approach with *permutation* of the individual words (word stems) and *filtering* of "interesting" N-grams, but their experiments only produced mixed results.

[9] The title of their study, "Syntactic Clustering of the Web", is rather misleading since no syntactic analysis is performed.

merged terms) to new bigrams as long as the *mutual information*[10] association measure between the two tokens is positive. Afterwards, the phrases thus found are used as a static repository to identify phrases in all further documents. New phrases cannot be found in that stage. Bakus *et al.* report on an improvement in clustering results compared to single words.

3.2.3.2 Syntactic Phrases

Sentences are made up of words, but the words cannot be arranged completely arbitrarily. Each language has a syntax which governs the arrangement of words. Thus, words are usually combined in groups (phrases) that fulfil specific functions in a sentence: noun phrases, verb phrases, prepositional phrases, adjective phrases, adverbial phrases. For an introduction to *grammar*, the formal description of a language, and *parsing*, the process of analysing a sentence and decomposing it into its constituents, see e. g. Jurafsky and Martin (2000, chapters 9–12).

Decomposing a sentence into its phrases intuitively promises a more accurate vector description than breaking it down to individual words. However, several IR experiments in earlier years have failed to show a substantial advantage of the more complex syntactic approaches if compared to statistical methods (Mitra *et al.*, 1997; Kraaij and Pohlmann, 1998; Sparck Jones, 1999). For document clustering the use of syntactic phrases (usually in addition to single words) has not yet been tested systematically. The results of studies in the document classification field are generally not too encouraging (see the review in Caropreso *et al.*, 2001). Nevertheless, Arampatzis *et al.* (2000a) report on good results with composite terms consisting of either adjacent word pairs (nouns and adjectives) or decomposed noun phrases.

Wen *et al.* (2001) mention the possibility of applying a noun phrase recogniser. In their actual work they preferred to rely on an existing phrase dictionary with a limited number of well-defined noun phrases. It seems questionable whether this method can be used in a more general clustering context. Nevertheless, keeping lists of standard complex expressions (e. g. "bond issue") are a worthwhile enhancement to the bag-of-words model (cf. Basili *et al.*, 2000).

Relatively well-explored is the topic of *named entity (proper noun) recognition*. Clifton *et al.* (2004) use such a module to identify persons, locations and organisations (see also Hotho *et al.*, 2002). While these studies replace BOW by named entities, Hatzivassiloglou *et al.* (2000) and Liu *et al.* (2002) suggest to combine BOW and named entities. See also Basili *et al.* (2000) whose text classifier identifies named entities as part of a more complex linguistic analysis.

Chu *et al.* (2003) discuss an approach using phrases from a domain-specific knowledge source (here: medical terms) to represent a document. In a suitable environment the use of such a phrase lexicon is bound to prove very useful.

3.2.3.3 Other Approaches

Two further methods which make use of context information are *lexical affinities* and *sentence clustering*.

Lexical Affinities. Maarek *et al.* (2000) use binary, alphabetically ordered *lexical affinities* instead of single words for clustering. Lexical affinities are collocations consisting of open-

[10] Mutual information between two tokens t_1 and t_2 is measured by

$$MI(t_1, t_2) = \log_2 \frac{P(t_1, t_2)}{P(t_1) \cdot P(t_2)}, \tag{3.3}$$

where $P(x)$ is the event probability of token x and $P(x, y)$ the joint probability of the two tokens x and y.

class lexical terms (nouns, adjectives, adverbs and verbs) occurring in a ±5 window of each other (Maarek and Smadja, 1989).

Sentence-Based Models. Using whole sentences as features is another obvious approach. Pullwitt and Der (2001) present a *double-clustering* method in which they first cluster all the sentences of all the documents into a fixed number of "sentence categories". Afterwards the documents are represented in terms of these categories rather than individual words. Given a sufficiently large document collection and documents of sufficient length, they report encouraging results.

Bellot and El-Bèze (2000) suggest to cluster the individual sentences in a tree structure. Afterwards they take the leaf nodes and replace all sentences by their original documents. The usefulness of the structure thus gained remains to be seen though.

3.2.4 Non-Textual Features

As has already been mentioned, hyperlinks can also be used for document representation, even though they are more common for graph models than for the vector space model (cf. Section 2.1.3). But exceptions exist such as the model of Modha and Spangler (2000) which represents each document by three separate sets of vectors: a word vector, an in-link vector and an out-link vector. The in- and out-link vectors show the links and the number of their occurrence between the present document and the set of all other documents meeting a certain minimal condition. The basic assumption is that documents sharing many in-linking or out-linking documents tend to be similar in content as well.

Other meta-data such as geographical location (Govindarajan and Ward, 1999), stylistic data and coefficients (Karlgren, 1999) or document *length*, *file type* and *creation/last update time stamps* may be used for clustering. While on their own these methods are hardly sufficient for a broader context, they may offer interesting additional possibilities to some of the more comprehensive approaches. In restricted environments more sophisticated meta-features could also be useful. For instance, in clustering scientific texts it might be beneficial to be able to differentiate between technical reports, conference proceedings, journal articles, etc.

3.3 Feature Weighting

It has long been customary in Information Retrieval to further modify the data gained by the vectorisation function in order to make the data better fit a particular model. One such modification is term or feature weighting.

In the BOW-model such term weighting schemes have led to substantial improvements of retrieval performance (Salton and McGill, 1983). Feature weighting has also been routinely used for clustering, but some authors express doubts as to its usefulness (e. g. Sneath and Sokal, 1973; Willett, 1988; Tombros *et al.*, 2002).

A feature weighting scheme w is an instance of a feature transformation function ϕ where $\mathcal{F}_1 = \mathcal{F}_2$ and it has three typical components (Salton and Buckley, 1988; Dhillon and Modha, 2001):

$$d'_{ij} = \phi(d_{ij}, H) = t(d_{ij}) \cdot g(d_{\cdot j}) \cdot s(d_{i\cdot}) \,, \tag{3.4}$$

where $d_{\cdot j}$ and $d_{i\cdot}$ stand for the respective row and column vectors. The individual components are

$t(d_{ij})$: a local feature weighting component,
$g(d_{\cdot j})$: a global feature weighting component,
$s(d_{i\cdot})$: a normalisation component.

Name	$t(d_{ij}) =$	Notes
none	d_{ij}	Simple "feature count"—the standard choice in most systems.
bin	$\mathrm{sgn}(d_{ij})$	Binary vectors—very rarely used these days. One relatively recent occurrence in clustering is the genetic algorithm of Jones *et al.* (1995). See also Lee *et al.* (2005) who claim better performance for similarity measuring with binary vectors.
maxtf	$0.5 + 0.5\frac{d_{ij}}{\max_l d_{il}}$	Augmented normalised term frequency (between 0.5 and 1) (Salton and Buckley, 1988). Very rare for clustering.
log	$\mathrm{sgn}(d_{ij}) \cdot \log(d_{ij})$	Logarithmic dampening is the second most popular option and also used relatively frequently. Modified versions such as $1 + \log(d_{ij})$ (e. g. Singhal *et al.*, 1996b) or $\log(1 + d_{ij})$ (Rüger and Gauch, 2000) are also common, with $t(d_{ij}) = 0$ if $d_{ij} = 0$ in each case.
sqrt	$\mathrm{sgn}(d_{ij}) \cdot \sqrt{\lvert d_{ij} \rvert}$	Square root dampening was used by Cutting *et al.* (1992). Larsen and Aone (1999) compare *sqrt*, *log* and *none*, finding little reason to use the former two.

Table 3.1: **Local feature weighting variants.** $\mathrm{sgn}(x) = -1, 0, 1$ depending on whether x is negative, zero or positive; furthermore we define $\log(0) = 0$.

3.3.1 Local Feature Weighting

Different local weighting schemes are described in Table 3.1. The various smoothing functions are motivated by very high values in certain columns, but in practice they are not too often used for document clustering.

3.3.2 Global Feature Weighting

The global weighting component is used to de-emphasise very frequent terms since they usually have little or no discriminative power. The common solution to do so is to multiply them by the logarithm of the so-called "inverse document frequency".[11] See Table 3.2 for the details.

"Global frequencies" usually mean those with regard to the document collection $\mathcal{S}$, but base frequencies from Ω would be preferable (though often unavailable).

3.3.3 Normalisation

Many IR systems use a normalisation function to eliminate differences in document/vector length as they are generally believed to have no influence on document quality or relevance. See Table 3.3 for the typical normalisation factors. An additional benefit in the Euclidean space is that

[11] "Inverse document frequency" is the accepted term, even though "inverse collection frequency" would be a more accurate description.

Name	$g(d_{\cdot j}) =$	Notes
none	1	Omit global weighting altogether.
idf	$\log \frac{n}{n_j}$	The classical "inverse document frequency" measure to de-emphasise frequent features.
idf-prob	$\log \frac{n-n_j}{n_j}$	A probabilistically motivated variant of the inverse document frequency, only rarely used in practice.
mi	$\log \frac{h_{ij}/N}{(\sum h_{i\cdot}/N)(\sum h_{\cdot j}/N)}$	The mutual information weighting scheme (Pantel and Lin, 2002) combines elements of local and global weighting. Despite a solid theoretical background it is only rarely used.

Table 3.2: **Global feature weighting variants.** Exceptionally, n_j is here re-defined as $n_j = \mathrm{df}(j, H)$ and the total feature sum $N = \sum h_{\cdot\cdot}$ (see also Salton and Buckley, 1988).

Name	$s(d_{i\cdot}) =$	Notes
none	1	Omit normalisation altogether.
L_1	$\frac{1}{\|t(d_{ij})\cdot g(d_{\cdot j})\|_1}$	Manhattan normalisation is occasionally used for document clustering.
L_2	$\frac{1}{\|t(d_{ij})\cdot g(d_{\cdot j})\|_2}$	Normalisation to Euclidean unit length is the standard procedure.

Table 3.3: **Vector length normalisation.** Shrinking vectors to unit length removes differences between long and short documents and makes it easier to compare them.

calculation of the cosine similarity measures can be reduced to the simple vector product:

$$\begin{aligned} s_{\mathrm{Cosine}}(\mathbf{d}_i, \mathbf{d}_k) &= \mathbf{d}_i^T \mathbf{d}_k \,, \\ \text{if } \|\mathbf{d}_i\|_2 = \|\mathbf{d}_k\|_2 &= 1 \,. \end{aligned} \tag{3.5}$$

In accordance with other IR applications most clustering algorithms use normalisation, even though in the mid-Nineties research has shown that in a retrieval context giving extra weight to longer documents might pay off (cf. Singhal *et al.*, 1996a,b).

In practice there is no limit to the number of different variations for feature weighting. For instance, Allan *et al.* (1997) use the following formula, which is also employed for clustering by Leuski (2001):

$$\hat{d}_{ij} = 0.4 + 0.6 \cdot \frac{d_{ij}}{d_{ij} + 0.5 + 1.5 \frac{\sum d_{i\cdot}}{\frac{1}{n}\sum d_{\cdot\cdot}}} \cdot \frac{\log \frac{n+0.5}{\mathrm{df}(j,H)}}{\log n + 1} \,. \tag{3.6}$$

For k-means clustering a re-sampling method was introduced by Modha and Spangler (2003) to determine an *optimal* feature weighting. However, it comes at a considerable computational cost and it is not yet clear under which circumstances the outcome justifies the means.

3.4 Feature Refinement

The bag-of-words and related document representations are often very high-dimensional and sparse, prompting the use of further refinement and reduction methods. A number of statistical reduction methods is thus known to bring down the size of large document-feature matrices. A second set of refinement methods derives from the fact that in document clustering the individual features are not just abstract and atomic entities, but may be further analysed on an orthographic, syntactic or semantic level, giving rise to a multitude of further representation methods.

In Section 1.6 the vector transformation function ϕ had been introduced, transforming a document vector from one feature space ($\mathcal{F}_1$) into another ($\mathcal{F}_2$):

$$\mathbf{d}'_i = \phi(\mathbf{d}_i, H), \quad \text{with} \quad \phi : \mathbf{R}^{m_{\mathcal{F}_1}} \rightarrow \mathbf{R}^{m_{\mathcal{F}_2}} . \tag{3.7}$$

Without aiming at too rigorous definitions, we can divide the feature refinement methods for illustrative purposes as follows:

Feature Selection. Methods that remove part of the features and leave the rest untouched. In other words, individual columns are removed from the matrix and the new feature space $\mathcal{F}_2$ is a subset of the original feature space: $\mathcal{F}_2 \subseteq \mathcal{F}_1$.

Feature Standardisation. Methods that bring the features (words) into standard forms. Some new features may have to be created and several old features may "collapse" onto one new feature. Essentially the features are still easily recognisable as words and the feature spaces bear a strong resemblance: $\mathcal{F}_2 \sim \mathcal{F}_1$.

Feature Extraction. Methods that define new, potentially very abstract features that are remote from the original features. The two feature spaces are thus no longer similar: $\mathcal{F}_2 \not\sim \mathcal{F}_1$.

Below follows a discussion of various such refinement methods. It may be noted that several but not all of them may be reasonably subjected to a linear combination.

3.4.1 Feature Selection

Feature selection (or feature *filtering*) aims to remove those columns from the document-feature matrix that contribute little or nothing towards explaining the desired properties of the data. The aim is twofold: to reduce "noise" and to decrease complexity.

Following work by Luhn (1958) it is generally assumed that words (lemmata, phrases, ...) with very high frequencies and those with very low frequencies do not contribute significantly to the content of a text. Several procedures have been developed to cut off features that are either too common or too rare. Figure 3.1 shows Luhn's hypothesis about the "resolving power" of features with regard to their frequency. Words to the right of D are too infrequent to convey substantial content but are less problematic than those to the left of C which occur all to frequently and threaten to bury the content by their "noise".

3.4.1.1 Stopword Removal

Probably the oldest feature refinement method in IR of all is stopword removal. Stopwords are loosely defined as frequent words carrying little or no useful information by themselves. Very typical cases are words such as "to", "be", "or", "not", etc.

Despite their ubiquitous application, stopwords are relatively unexplored. Only little work has gone into the systematic determination of stopwords (Fox, 1992; Wilbur and Sirotkin, 1992;

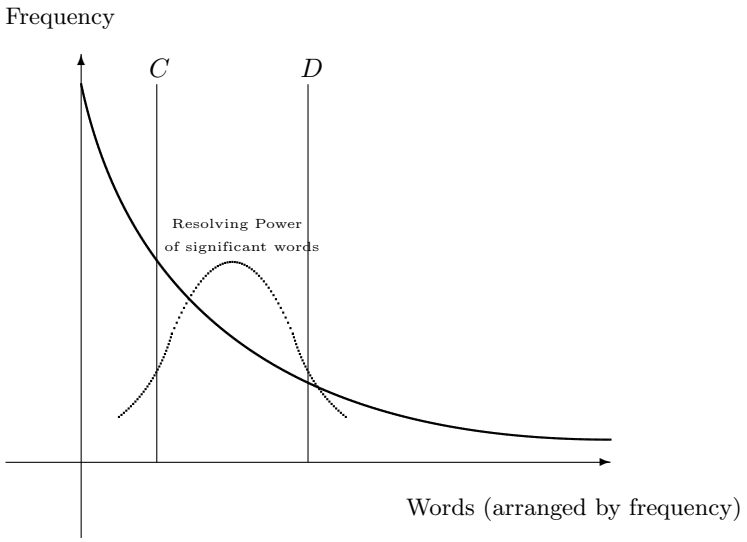


Figure 3.1: **Luhn's curve.** Resolving power (dotted line) of words with regard to their frequency (straight line) (Luhn, 1958). C and D are upper and lower cut-off points. Words bearing real content normally lie in the range between those two.

a	alone	another	be	being	cannot	enough
about	along	any	became	below	co	etc
above	already	anyhow	because	beside	could	even
across	also	anyone	become	besides	down	ever
after	although	anything	becomes	between	during	every
afterwards	always	anywhere	becoming	beyond	each	everyone
again	among	are	been	both	eg	everything
against	amongst	around	before	but	either	everywhere
all	an	as	beforehand	by	else	except
almost	and	at	behind	can	elsewhere	few

Table 3.4: **Stoplist.** The first 70 of the 250 terms on the popular stoplist devised by van Rijsbergen (1979).

Sinka and Corne, 2003a,b). Most applications rely on the old list provided by van Rijsbergen (1979, see Table 3.4), on the one derived from the Brown corpus (Fox, 1992) or on ad-hoc creations. For clustering in a particular domain, a specific stoplist should be used. For instance, for clustering sport reports, terms such as "winner" and "loser" carry no discriminative power and should be omitted lest they actually obstruct clustering.

Stoplists for search engines should usually be enlarged by typical Web terms such as "home", "back", "contact", "webmaster", etc. which usually add nothing at all to distinguish between the content of different documents.

Even though relatively little attention is paid to stoplists, their usefulness is almost universally accepted and there is rarely an IR application that does not benefit from a careful application of stopword removal.

3.4.1.2 Pruning

We use the term "pruning" for all those methods that reduce the feature space not by looking at the properties of individual features but by looking at the feature matrix and the frequencies of the features. The aim is again twofold: first of all to discard features that have little or no discriminative power, i. e. those at the edges of Luhn's curve (though pruning is mainly addressing the infrequent features and not the very frequent ones). Secondly, the goal is to reduce the size of the matrix to the most significant dimensions and thus speed up similarity calculations.

Local Pruning deals with each document vector individually and without regard to the other documents. Typically, of each vector the q most frequent features are retained, though a concentration on medium-frequency features could also be imagined.

Weiss *et al.* (2000a,b) use an approach where each document is represented by the words in its title and in other special keyword tags as well as the q *most frequent terms* in the text body (after stopword removal), with q set to a low value (e. g. $q = 8$). Similar procedures are described by Cutting *et al.* (1993), who use the 50 highest weighted features of each vector, and by Schütze and Silverstein (1997), who perform successful experiments with $q = 20$ and $q = 50$. In each document the values for the less frequent words are simply set to zero. In a different setting Pantel and Lin (2002) also fared well with a local reduction technique. On the other hand, Larsen and Aone (1999) find that $q = 25$ leads to acceptable but worse results than $q = 250$.

Local pruning does not necessarily lead to an overall reduction of dimensions. Still, substantial savings in computational cost are possible because of the reduction of non-zero matrix elements to a fraction of the original number.

Global Pruning looks at the overall feature frequencies to determine the final selection. The clustering engine of Hannappel *et al.* (1999) simply restricts document representation to the q most frequent terms in the collection (after stemming and stopword removal); from these representations the engine then distills so-called topics which are used for cluster description. Bradley and Fayyad (1998) use the top 302 words (without stemming or stopword removal). Related approaches are discussed by Yang and Pedersen (1997) for text classification.

Dhillon and Modha (2001) eliminate non-content-bearing "high-frequency" and "low-frequency" words, while the search result clustering algorithm of Rüger and Gauch (2000) uses a special weighting scheme to determine the most useful terms:

$$w(f_j) \;=\; \frac{\mathrm{df}(j, H_{\mathcal{S}})}{\mathrm{df}(j, H_{\Omega})} \quad \cdot \quad \mathrm{df}(j, H_{\mathcal{S}}) \log \frac{n}{\mathrm{df}(j, H_{\mathcal{S}})}. \tag{3.8}$$

Only the q features with the highest weights are kept. The first factor favours words that are specific to the selected documents (compared to the document universe Ω), while the second factor favours terms with a *medium document frequency*. Rüger and Gauch report that for $n = 1000$ and q between 9 and 11 the clustering quality was significantly better than for the full set of features.[12]

Volk and Stepanov (2001) present two sampling methods by which they iteratively find highly discriminative word subsets via an entropy threshold. However, the more effective such method, *word set re-sampling*, comes with prohibitively high computational costs of the order $O(n^3)$.

3.4.1.3 POS Selection

A linguistically motivated more general alternative to stopword filtering is feature selection based on part-of-speech tags. For instance, Basili *et al.* (2000) consider just three "open" word-classes (nouns, verbs and adjectives)[13] for their classification tasks while Rüger and Gauch (2000) rely only on nouns. On the other hand, a more detailed POS indexing scheme is described by Arampatzis *et al.* (2000a,b). They report encouraging results in the field of text classification, which makes their approach appear promising for text clustering as well.

Henderson *et al.* (2002a,b) use syntax analysis to select just those terms that function as heads in noun or verb phrases, leading to a reduced document representation while still providing acceptable results. Attempts to further restrict the heads to noun phrases occurring in subject or object position (with regard to the main verb phrase) turned out to be too severe, leading to a significant reduction in clustering quality.

[12] Prior to applying their weighting scheme, Rüger and Gauch already reduce the feature space to medium-frequency nouns (i. e. nouns f_j with $\frac{1}{3}\mathrm{df}(j, H_{\Omega}) \geq \mathrm{df}(j, H_{\mathcal{S}}) \geq 3$). The method obviously requires sufficient knowledge about the document universe.

[13] *Open* word classes are those that have an unlimited number of members, with new creations (through compounding, derivation, invention, borrowing, etc.) being constantly added. A typical open word class is the noun class. Closed word-classes are much smaller and more stable, new members are only added in exceptional cases. A typical closed word class is the pronoun class.

3.4.1.4 Advanced Weighting

The feature selection methods just described can be interpreted as a *binary feature weighting* procedure. The restriction that the weights have to be either 0 or 1 can, of course, be lifted. As a result a large number of individual weighting schemes can be imagined, either as additions to or as substitutions for the general weighting schemes discussed in Section 3.3.

Lending extra weight to selected terms (e. g. title tokens or proper names) is a popular means of stressing some "relevant" features without risking the accidental loss of some other important feature. For instance, Hatzivassiloglou *et al.* (2000) identify noun phrase heads and proper names and give them extra weight.

3.4.2 Feature Standardisation

Features, and in particular words, can be standardised in numerous ways. The aim of such standardisation procedures is to collate features that are in some respect very similar to each other. Noise generators such as spelling, style or other linguistic preferences unrelated to the content of the document should thereby be eliminated.

Formally speaking, feature standardisation is defined by an $m' \times m$ weighting matrix W:

$$\mathbf{d}' = \phi(\mathbf{d}, H) = W\mathbf{d}\,. \tag{3.9}$$

Row $W_{\cdot j}$ is a mapping vector for the original feature f_j. For most standardisation methods the new feature set $\mathcal{F}'$ is smaller than the old set $\mathcal{F}$ and each original feature is mapped onto exactly one new feature:

$$m' \ll m\,, \tag{3.10}$$

$$W_{\cdot j} = (0\ldots0, 1, 0\ldots0)^T\,. \tag{3.11}$$

The simplest and most universal standardisation method is mapping all features to lower case spelling ("case folding"). A discussion of several more sophisticated mappings follows in the sub-sections.

3.4.2.1 Truncation and Stemming

Truncation and stemming are simple methods aiming to remove *word suffixes* which lead to a diversification of words that could otherwise be regarded as principally identical.

Truncation: Mechanically chopping off (from the end) or retaining a fixed number of letters of each word. This archaic reduction technique is hardly ever used in document clustering nowadays since the resulting collations are too arbitrary.

Stemming: Algorithmically reducing words to an (often) artificial word "stem" (not to be confused with lexical stems). Stemming rules are usually quite simply formulated and do not require any real linguistic (morphological) knowledge. Porter's extremely popular algorithm for English language stemming (Porter, 1980, see Table 3.5 for some examples[14]) has found wide adaption in IR in general and document clustering in particular. However, some studies such as Riloff (1995) as well as Sinka and Corne (2002) cast doubt on the universal usefulness of stemming.[15]

[14] See also `www.tartarus.org/~martin/PorterStemmer/` and `snowball.tartarus.org`.

[15] For an approach to stemming which is based on the *clustering of words* see Baeza-Yates *et al.* (2003).

Fried	⟶	fri
oscillating	⟶	oscil
eggs	⟶	egg
puzzled	⟶	puzzl
Woody	⟶	woodi
comfortably	⟶	comfort
into	⟶	into
critically	⟶	critic
enhanced	⟶	enhanc
compulsive	⟶	compuls
playfulness.	⟶	play.

Table 3.5: **Stemming example.** Porter's original stemming algorithm applied to a not entirely random sentence.

3.4.2.2 Lemmatising

Lemmatising is the linguistic technique to reduce words to their lexical base forms which is particularly useful for inflecting languages. Since word forms can often belong to more than one lexical stem (e. g. "can" as a noun or verb), word forms disambiguated with POS tags are a preferable input than bare word forms. Using both lexical and morphological knowledge, so-called lemmatisers can then transform word forms into the original lemmata.

It is often believed that lemmatising brings only few additional benefits compared to the more aggressive stemming while requiring significant extra effort. It is therefore used relatively rarely (cf. Smeaton, 1997). One study using lemmatising is the work by Henderson *et al.* (2002a,b).

3.4.2.3 Compound Splitting

Some languages such as German and Swedish allow multiple words to be joined together in various ways so that they form new words. These languages are rich in *compounds*. Compound words such as Mark Twain's famous "Personaleinkommensteuerschätzungskommissionsmitgliedsreisekostenrechnungsergänzungsrevisionsfund" obviously contain a whole lot of information about individual concepts (Personal–Einkommen–Steuer–Schätzung–Kommission–Mitglied–Reise–Kosten–Rechnung–Ergänzung–Revision–Fund) which are lost if the word is just regarded as one big black box or if only its suffix is normalised by stemming or lemmatising.

With clustering experiments in languages other than English still being relatively rare, not much study has yet gone into compound splitting techniques. One exception is the work by Rosell (2003), but a number of questions relating to appropriate splitting points and weights still require further investigation. For instance, compounds that have long been in use and become part of the established vocabulary (e. g. "Strohfeuer") should normally *not* be split.

Unlike for the previously discussed techniques, the weight vector $W_{\cdot j}$ for a compound has usually not just one but several non-zero positions, meaning that a feature from the original representation can be reflected in more than one position of the new document vector.[16]

3.4.2.4 Semantic Concepts

The ultimate goal of most clustering efforts is not to bring together documents with similar words or word forms, but documents with related *content*. It is therefore natural to look out for approaches that represent documents by their *semantic* concepts. The most ambitious such attempt for clustering has been proposed by Choudhary and Bhattacharyya (2002) who want to represent documents in *Universal Networking Language* (Uchida *et al.*, 1999). The UNL language models sentences as graphs, with "universal words" as nodes and semantic relations as edges. Each document could then be represented by its universal words, with the number of connecting edges being the frequency d_{ij} of each word. At present, however, the translation of natural language texts into UNL is still illusionary.

More realistic approaches use existing ontologies. For example, Clifton *et al.* (2004) present an approach based on WordNet (Miller *et al.*, 1990) to expand the keyword list and catch synonym as well as hyper-/hyponym relations between terms. Based on their initial experience they define rules when to apply WordNet information.

Work by Chu *et al.* (2003) uses a medical knowledge source with hierarchically organised concepts (similar to WordNet). They describe a sophisticated formula to exploit the knowledge built into the hierarchy. Essentially, the similarity between two documents is the sum of pairwise similarities between all their features (concepts). The contribution of a feature pair (f_1, f_2) with f_1 from D_1 and f_2 from D_2 is 1 if the two are identical, but otherwise it is calculated by taking three contributing factors into account: (1) "hopping distance" in the concept hierarchy between f_1 and f_2; (2) generality of f_1 and f_2 (the higher up in the hierarchy, the smaller the respective similarity), and (3) orthographic similarity of the word stems of f_1 and f_2 which is used as a precaution against missing links in the concept hierarchy.

A domain-specific ontology is also used by Hotho *et al.* (2002) to achieve both generalisation and feature reduction (in German). Their algorithm projects the words of the text onto concepts of the ontology, keeping and expanding the concepts that are sufficiently "supported" by the text. Each document is thus represented by a limited number of well-chosen concepts, speeding up clustering and increasing accuracy.

Ontologies are attractive because they allow mapping of orthographically very different but semantically close concepts onto a single feature. However, in practice the application is not easy because words are often *ambiguous* and have multiple meanings (Gonzalo *et al.*, 1998). Feature representation through semantic concepts thus often suffers from over-generalisation: as a result of spurious connections between individual words, some documents are suddenly regarded as related even though they may have nothing at all in common. Often additional *word-sense disambiguation* analysis is necessary to achieve a reliable representation (cf. Schütze, 1998; Leacock *et al.*, 1998; Stevenson, 2003); taking the most frequent sense has often been preferred as a less costly alternative (Chu *et al.*, 2003).

3.4.3 Feature Extraction

Feature extraction methods break out of the ordinary BOW feature frame. Starting from the existing document features they build new and sometimes totally abstract features that can no longer be easily determined by the eye or by looking at an individual document. Most of the

[16] But not always (e. g. "Sohnessohn"!).

methods in this section perform transformations on the entire document-feature matrix and not just on individual rows or columns. Another recurring element is the goal to bundle features that are thought to convey similar meanings.

3.4.3.1 Double Clustering

Clustering being a way to bring order into a large number of objects, it is only natural that more than once it has been suggested to use clustering not just for the documents but also for the features. Such *double clustering* approaches either cluster sequentially first features and then documents based on these new features or both simultaneously.

Frequent Itemsets. *Association rule mining* (Agrawal *et al.*, 1993) can be used to establish sets of frequently co-occurring terms. Several specific algorithms have then been suggested for clustering using such *frequent itemsets*. A graph-based method was briefly described in Section 2.1.3, other algorithms were developed by Beil *et al.* (2002) and Fung (2002).

Information Bottleneck Method. A statistical approach is used by Slonim and Tishby (2000). Their double-clustering method uses the *information bottleneck method* (Tishby *et al.*, 1999) to achieve a significant reduction in dimensionality with minimal loss of information. Based on a *mutual information measure*, words are first clustered hierarchically such that the new features (word clusters) maintain as much information on the documents as possible. In a second step the documents, represented on the basis of these word clusters, are themselves clustered hierarchically. The algorithm has performed well in comparison to several standard techniques but is computationally more expensive, having a complexity of $O(n^3)$. In later work the *sequential information bottleneck* method was presented, with reduced time and storage complexities (Slonim *et al.*, 2002).

For *word clustering methods* see also Pereira *et al.* (1993) and Dhillon *et al.* (2002) and for clustering based on sentence clusters Pullwitt and Der (2001, cf. also Section 3.2.3.3). *Adaptive Subspace Iteration* (Li *et al.*, 2004) is another recent iterative algorithm working on a two-step principle.

3.4.3.2 Latent Semantic Analysis

Inspired by other work in statistics, various attempts have been made at tackling the document clustering problem with algebraic methods. By far the most important such method is *Latent Semantic Analysis (LSA)/Latent Semantic Indexing (LSI)* (Deerwester *et al.*, 1990).

Based on the assumption that there is an underlying semantic structure in the data, which is partially obscured by "noise" (randomness of word distribution), latent semantic indexing aims to reduce the high-dimensional, redundant word space to a low-dimensional, orthonormal "concept" space. *Singular Value Decomposition* (SVD, cf. Golub and van Loan, 1996, 70–71) is applied to the document × term matrix. By ordering the dimensions and retaining just those N dimensions corresponding to the highest eigenvalues, the best possible N-dimensional approximation to the original matrix is obtained (i. e. the approximation with the lowest least-square distance to the original).

The method was originally devised for retrieval purposes, but experiments have shown that in clustering it also permits great dimensionality reductions at a comparatively small price in quality. Schütze and Silverstein (1997) show that LSA-reduction from several thousand to 20, 50 or 150 dimensions results in clustering solutions on par with those in the full term space. A similar result was obtained by Hasan and Matsumoto (1999) with 30 dimensions. Lerman (1999) discusses the question of finding the optimal number of dimensions. In a small collection of 1000

documents she finds that as few as six dimensions produce the best results, but for larger and more diverse collections the optimal number is expected to be higher. Kanejiya *et al.* (2004) show how to use LSA with syntactically or semantically annotated word tokens.

LSA generally reduces the number of dimension very drastically (and results in very compact matrices instead of the sparse ones usually encountered in document clustering). The drawback of Latent Semantic Analysis is that SVD is of complexity $O(n^3)$ and even the approximations are computationally very expensive.

3.4.3.3 Random Indexing

Random indexing (Kanerva *et al.*, 2000; Sahlgren and Cöster, 2004) is a recent method aiming to significantly reduce matrix size and at the same time avoiding to have to perform costly calculations on the entire matrix ("the huge matrix step"). The principal idea is to replace the truly orthogonal feature dimensions by *nearly* orthogonal vectors. In high-dimensional spaces the number of "nearly orthogonal" vectors is much higher than that of the truly orthogonal vectors and the data can thus be represented in a lower dimensionality without much loss of information. Moreover the approach is incremental and does not require the whole matrix to be known in advance.

3.5 Time Constraints

The feature vectorisation, weighting and refinement methods that have been discussed in Sections 3.2 to 3.4 present a large mix of techniques, each of which comes with its specific advantages and disadvantages. In particular, much time is often spent to produce representations that are more accurate or more compact. In the present section we will attempt to provide an overview of the suitability of these methods for different clustering tasks depending on their time constraints.

We suggest to distinguish three stages of a clustering application:

Vectorisation: Storing and vectorising the documents plus all feature weighting and refinement operations that can be performed for each document individually.

Matrix assembly: Gathering the document vectors into a matrix and performing all matrix-dependent weighting and mapping operations.

Clustering: The actual clustering phase.

Table 3.6 shows a scheme for classifying clustering applications with regard to the time-criticality of these three phases and introduces four scenarios with different requirements: off-line clustering, repeated clustering, ad-hoc clustering and instant clustering.[17]

Table 3.7 in turn shows which methods occur at which stages, providing a first rough guideline towards which techniques to use for which kind of application.

If the actual clustering is performed on-line, fast clustering algorithms are necessary. Suffix tree clustering, hierarchical divisive clustering and some of the partitional methods appear to be the methods of choice. Hierarchical agglomerative, probabilistic and double-clustering methods appear to be better suited if speed is less critical.

[17] Those situations where the documents arrive one-by-one and clustering is done in parallel with document processing (*incremental clustering*, cf. Section 2.3.2.1) belong probably all to the instant-clustering scenario.

	Vectorisation stage	Matrix assembly stage	Clustering stage
Off-line clustering. Applications with no time-critical elements such as pre-clustering a document collection for cluster-based retrieval.			
Repeated clustering. Applications which perform different cluster applications on a constant matrix. Typically, one or several clustering algorithms are repeatedly run with different parameters. Such parameters could be set by a user who thus is given some control of the clustering interface.			x
Ad-hoc clustering. Applications without a predetermined document set. Both the selection of documents and clustering are performed on-line (e. g. clustering of search engine results).		x	x
Instant clustering. Applications which cannot rely on predetermined vector representations. They start with raw documents which must be analysed on the fly. An example is a clustering interface to a meta search engine.	x	x	x

Table 3.6: **Cluster scenarios.** A basic scheme of clustering applications with regard to their time-critical components.

Section	Method	Vectorisation stage	Matrix assembly stage	Clustering stage
	Document Vectorisation			
3.2.1	Simple document tokenisation (BOW)	x		
3.2.2	Vectorisation with linguistic annotations (POS tagging, sentence-parsing, etc.)	X		
3.2.3.1	Statistical phrase extraction	x		
3.2.3.2	Syntactic phrase extraction	x		
3.2.3.1	Suffix tree construction		x	
	Feature Weighting			
3.3.1	Local feature weighting	x		
3.3.2	Global feature weighting		x	
3.3.3	Vector normalisation (depends on previous feature representation methods selected)	(x)	(x)	
	Feature Refinement			
3.4.1.1	Stopword removal	x		
3.4.1.2	Local pruning	x		
3.4.1.2	Global pruning		x	
3.4.1.2	Global pruning with re-sampling		X	
3.4.1.3	POS Selection (requires prior POS tagging)	x		
3.4.1.4	Various advanced weighting schemes	x		
3.4.2.1	Truncation and stemming	x		
3.4.2.2	Lemmatising/morphological analysis	X		
3.4.2.3	Compound splitting (depending on methods)	(X)	(X)	
3.4.2.4	Semantic mapping (and word sense disambiguation)	X		
3.4.3.1	Sequential double-clustering		X	
3.4.3.1	Simultaneous double-clustering			X
3.4.3.2	Latent semantic analysis		X	
3.4.3.3	Random indexing	x		

Table 3.7: **Representation methods and application stages.** Different document representation and clustering methods occurring at different stages of a clustering application. X indicates major, x minor computational costs.

3.6 Cluster Presentation

Many real-world applications require a way of presenting the final cluster solution to the end-user in a practical and informative fashion. Below we discuss several aspects of cluster presentation: visualising the cluster structure (Section 3.6.1), describing individual clusters (Section 3.6.2) and interaction between user and system (Section 3.6.3).

3.6.1 Clustering Structure

The traditional solution for displaying clusters is text-based and on the surface similar to the well-known *Open Directory Project*[18], which is a hierarchically structured collection of manually selected Web pages. At each step of his travel through the directory the user sees one particular node of the hierarchy: first a list of all sub-categories, followed by a list of directly relevant document links (see Figure 3.2).

Some applications display an expandable and reducible hierarchy in a separate frame, giving the user a better overview of the overall structure and allowing the popular "Explorer" browsing strategy. See, for instance, the interface of the Vivísimo meta search engine (Figure 3.3). An alternative is to display the hierarchy in the main window and list the top one or two documents under each entry. Only upon the user's explicit choice of a certain cluster all entries are displayed (e. g. Maarek *et al.*, 2000). A "dynamic" explorer structure is described by Osdin *et al.* (2002).

More advanced two- and three-dimensional graphical representations are discussed by Hearst (1999b,c). A clustering algorithm producing *eo ipso* a two-dimensional mapping is WEBSOM (Section 2.3.2). It is not yet established how useful such sophisticated spatial arrangements are for the end-user or whether the customary text-oriented models are not actually preferable.

A minor problem with browsable hierarchies is that the binary structure created by a standard hierarchical algorithm is usually not very desirable. However, there are several ways of solving the problem easily; for example, by displaying the eight great-grandchildren of every node instead of just the two immediate children nodes. Other approaches use non-binary trees (Maarek *et al.*, 2000) which also solve the problem handily. The Scatter/Gather algorithm (Cutting *et al.*, 1992, 1993) is based on a different idea: Initially, the documents are divided into a flat partition ("scattered"). The user then selects one or several clusters; these are collected ("gathered") and again scattered into several sub-groups. Thus, the hierarchy is only developed "on-the-fly" as demand arises.

3.6.2 Cluster Description

Probably even more important for the success or failure of a clustering application than visualisation of the structure is an appropriate description or summary of the individual clusters ("cluster annotation").

Below follow brief descriptions of various elements that have been used for cluster annotation. Clearly, there is always a trade-off between the amount of information provided and the clearness and simplicity of the presentation.

Keywords/Key Features (*Cluster Digest*). The basic description of a cluster is almost always derived from its most typical clustering features. These may be the features with the highest weights on the cluster centroid, but special weighting schemes (including *tf-idf*) are also applied and may lead to better results.

[18] `www.dmoz.org`

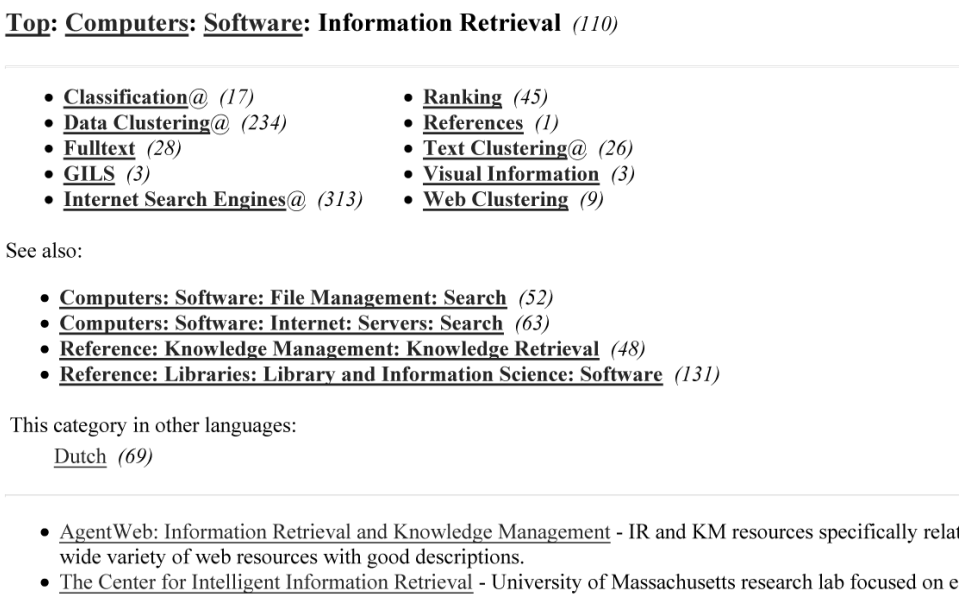


Figure 3.2: **Open Directory Project.** Typical view of the Open Directory hierarchy with sub-classes and cross-references on top, followed by individual documents.

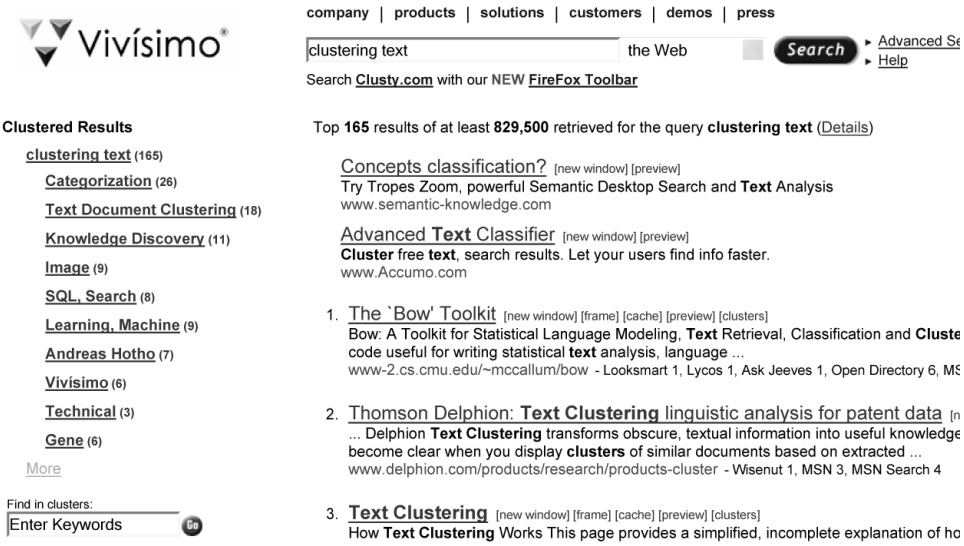


Figure 3.3: **Vivísimo**. Clustering interface to the Vivísimo meta search engine (`www.vivisimo.com`), with clusters on the left and individual document references on the right.

For instance, Chang and Hsu (1998) suggest to use for each cluster C_j the features (terms) f with what they call the highest *normalised cue validity* (*ncv*):

$$ncv(f_j, C_i) = \frac{p(f_j, C_i)}{p(f_j, C_i) + p(f_j, \bar{C}_i) + \varepsilon} \cdot \log \frac{1}{p(f_j, \Omega)}, \tag{3.12}$$

where the individual *feature sum* of feature f_j in document set A is $\mathrm{fs}(j, A) = \sum d_{ij} \wedge \mathbf{d_i} = \phi(\tau(D_i \in A))$, the cluster complement $\bar{C}_j = \mathcal{S} \setminus C_j$ and the term probability $p(f_j, C_i) = \mathrm{fs}(j, C_i) / \sum_{q=1}^{m} \mathrm{fs}(q, C_i)$. ε is a very small value ensuring that the first term is smaller than one.

Cluster digests derived from phrases (Zamir and Etzioni, 1999; Hannappel *et al.*, 1999) or named entities (Clifton *et al.*, 2004) are considered more humanly-understandable than those derived from simple bag-of-words models, which may result in some rather weird or "ugly" descriptions, especially if word-stemming had been applied in a previous processing step. Transforming the stemmed forms into nouns by lexical look-up may provide a solution. For a discussion of cluster description by phrases and sentences see Silva *et al.* (2001).

Size. Indicating the number of documents in a cluster is a simple but often helpful information. The number of sub-clusters may also be indicated in a hierarchical setting.

Cluster Quality. Chang and Hsu (1998) also display the average group similarity of each cluster as a measure of cohesion. Other compactness and separation measures are also used to indicate the relative "quality" of each cluster.

Representative(s). A popular method is to describe a cluster by giving title and link of its most representative document(s). In Chang and Hsu (1998) a document description consisting of the first few lines of text is also given. Several methods for choosing these representatives have been used. Chang and Hsu consider those documents nearest to the cluster centroid and choose the one with the least *URL depth* (as measured by the number of slashes). Leuski (2001) finds that in a search engine experiment the highest ranked document in a cluster is a better representative than the one nearest to the centre.

Sentences. Osdin *et al.* (2002) suggest to pick descriptive sentences from the documents in each cluster. The sentences are chosen that best match the original query terms. The result is similar to the popular document "snippets" returned by most of the current search engines.

Well-Linked Sites. In addition to a representative document, Modha and Spangler (2000) take link structure into account to select four more document titles/links for describing the cluster (called "breakthrough", "review", "citations" and "references"). These are cluster documents that show the most typical "in-link" and "out-link" patterns and those with most in- and out-links from and to the cluster (in relative terms).

Clearly, the careful selection of key words/features is central to the success of most document clustering applications (Stefanowski and Weiss, 2003). This has also been recognised by the commercial clustering application *Vivísimo* which already in the clustering process discards clusters which cannot be described concisely, accurately and distinctively (Vivísimo, 2003). Details of their algorithm are not disclosed, though the work by Pericliev and Valdés-Pérez (1998) and Palmer *et al.* (2001) gives an idea of how the search for clear *conceptual* descriptions may work.

3.6.3 Interactive Clustering

Only relatively little research has been devoted so far to exploring interactive aspects of document clustering. However, as stressed by Kreulen *et al.* (2001) and Spangler and Kreulen (2002) data mining (and document clustering) should no longer be regarded as a "hands-off" task. Instead, user feedback can play a critical role and we expect to see more research in that direction. Spangler and Kreulen describe an approach which is ultimately geared towards a classification scheme that is created by an interactive process of clustering and human intervention.

A more traditional form of interaction is the relevance feedback used by Leuski (2001) to dynamically re-order documents within a cluster, so that those are listed on top which are deemed most representative with regard to the user's prior browsing.

While the experiments presented in this study will focus on technical aspects of document representation, it is felt that practical issues and in particular user interaction are important topics for future research.

Chapter 4

Experimental Setup

Then of the two propositions,
both of them Aristotelian doctrines,
the second—which says it is necessary
to prefer the senses over arguments—
is a more solid and definite doctrine than the other,
which holds the heavens to be inalterable.
Therefore it is better Aristotelian philosophy
to say "Heaven is alterable
because my senses tell me so,"
than to say, "Heaven is inalterable
because Aristotle was so persuaded by reasoning."

Galileo Galilei in *Dialogue Concerning the Two Chief World Systems* (1632)

It is the goal of this thesis to increase empirical knowledge about the usefulness of natural language processing techniques in a specific setting: document representations of German-language texts for clustering. The present chapter describes the experimental setup consisting of the document data (Section 4.1), the clustering software (Section 4.2) and the evaluation methodology (Section 4.3). Finally, there follows a description of a few preliminary experiments that were conducted to determine the central parameters of the clustering software (Section 4.4).

Following common practice in document clustering evaluation, we chose a setup with *labelled* (i. e. *categorised*) documents. These labels are not known to the algorithm. A cluster solution is evaluated by identifying the overlap/similarity of the clusters and the categories (see Sections 2.6.2 and 4.3).

4.1 Document Data

In the absence of suitable reference collections for clustering experiments in German, we constructed our own database. Following a brief review of some commonly-used English data sets, we present our German ones in some detail.

alt.atheism	rec.sport.hockey
comp.graphics	sci.crypt
comp.os.ms-windows.misc	sci.electronics
comp.sys.ibm.pc.hardware	sci.med
comp.sys.mac.hardware	sci.space
comp.windows.x	soc.religion.christian
misc.forsale	talk.politics.guns
rec.autos	talk.politics.mideast
rec.motorcycles	talk.politics.misc
rec.sport.baseball	talk.religion.misc

Table 4.1: **The *20 Newsgroups* data set.**

4.1.1 Brief Review of English Collections

Experiments with clustering algorithms have been performed on a large variety of test collections. Many experiments were conducted on data sets generated ad-hoc, making it impossible to compare results across different studies. The following list names some of the more widely used *standard* test sets for English.

Reuters Test Collection. In recent years the probably most popular test collection for text clustering, categorisation and a number of other tasks has been the *Reuters-21578* corpus (Lewis, 1997; Hettich and Bay, 1999). Reuters-21578 is a collection of 21,578 articles which had appeared on the Reuters newswire in 1987. Each article has been manually indexed with a variable number of topics (135), people (267), places (175), organisations (56) and (stock) exchanges (39). The whole collection was formatted in SGML and has a size of 28 MB (uncompressed). The present polished version has been available since 1997.[1]

Reuters Corpus Volume 1 (RCV 1). The sequel to the popular Reuters-21578 collection is the *Reuters Corpus Volume 1* (Rose *et al.*, 2002), which contains 806,791 English language articles collected between 20 August 1996 and 19 August 1997. The collection is formatted in XML and has a size of 3.7 GB. The articles are tagged with one or more topics (103) and region codes (366) as well as zero or more industry codes (376). Topics and industry codes are part of an hierarchical structure.

20 Newsgroups. Another popular test collection has been the *20 Newsgroups (20NG)* set (Lang, 1995; Hettich and Bay, 1999), a collection of nearly 20,000 Usenet messages evenly distributed over 20 newsgroups. Some of these groups are very similar to each other, others distinctly different from the rest (see Table 4.1). The data set has an uncompressed size of 61.6 MB.

OHSUMED Collection. With 348,566 labelled titles/abstracts of medical articles from the period 1987–1991 this is one of the popular test sets regularly used at the annual **Text REtrieval Conferences (TREC)**.[2]

[1] An earlier version was Reuters-22173.
[2] `http://trec.nist.gov`

Associated Theme	Category
Banking & Finance	Commercial Banks Building Societies Insurance Agencies
Programming Languages	Java C/C++ Visual Basic
Science	Astronomy Biology
Sport	Soccer Motor Sport Sport (without soccer and motor sport)

Table 4.2: **The *BankSearch Dataset.*** Each class contains 1,000 documents.

Web Page Collections. Modern document clustering studies are very often geared towards the clustering of Web pages. Standard test collections have been unavailable for a long time so that each researcher has had to construct his own test set. In 1997 a separate TREC *Web Track*[3] (initially "Very Large Collection Track") was started, which resulted in two large standard collections of Web documents: *WT100g (VLC2)* (100 GB; over 18 million documents, based on an Internet Archive crawl in 1997) and *.GOV* (18.1 GB; ca. 1.25 million documents from the .gov domain, 2002). These pages are not indexed, however, and relevance judgements exist only for documents and topics used in past TREC tasks (Craswell and Hawking, 2002).

Various researchers have made use of the Open Directory Project[4] or the Yahoo! Directory[5] to gather labelled Web document collections. Sinka and Corne (2002, 2004) give an overview of such endeavours and provide the **BankSearch Web Document Dataset** as a large benchmark for Web document clustering.[6] The collection consists of 11,000 hypertext documents belonging to eleven more and less diverse categories which are part of four different associated themes (see Table 4.2).

At present it cannot yet be judged whether Sinka and Corne's data set will gain wider acceptance as a reference database.

4.1.2 Five German Data Sets

In the absence of comparable standard data collections for German texts, we decided to build a repository of our own. The following considerations played a role:

- Number of data sets: in order to reach a higher generality, there should be not just one but several entirely independent corpora drawn from different sources.
- Size: our corpora should be sufficiently large; the larger the better.

[3] `es.cmis.csiro.au/TRECWeb`
[4] `www.dmoz.org`
[5] `www.yahoo.com`
[6] Used to be available until summer 2005 from `www.pedal.reading.ac.uk/banksearchdataset/`.

- Number of categories: at least some of the corpora should feature more than just a handful of categories.

- Uneven distribution: the categories should have different sizes, making the clustering task both harder and more realistic.

- Length: for similar reasons, documents should have different lengths. Moreover, in order to reduce random effects they should not be too short.

HTML constructs and other tags were stripped from all documents which were then stored in an XML format. Below we describe the five data sets that have been gathered individually, concluding with a summary of the key properties of all five. In all instances documents that were too short (less than twenty tokens, less than fifteen types or less than five nouns) were excluded. It was also attempted to remove all documents that were recognised as non-German. Duplicates were eliminated as well as possible.[7]

4.1.2.1 Springer Data Set

Name	SPRINGER
Source	A collection of German book descriptions found at `www.springeronline.com` in December 2004.
Categories	The German Springer titles come in 28 different categories. Of these the seven largest were retained. See Figure 4.1.
Number of Categories	7
Number of Documents	3,836
Documents per Category	258–1,148 (average: 548)
Document Length (Tokens[8])	20–269 (average: 88)
Document Length (Types)	15–185 (average: 70)
Tokens in Corpus	335,658
Types in Corpus	46,232
Token Length	2–63 letters (average: 7.24)

[7]The entire *data cleansing* problem was treated rather in an ad-hoc manner in this study. See for example Sung *et al.* (2003) for a more careful and detailed examination which, incidentally, uses clustering techniques for data cleansing.

[8]In the subsequent descriptions, "document length in word *tokens*" is counted as all individual occurrences of text units in a document with at least two letters. "Document length in word *types*" is the number of *different* such tokens in each document.

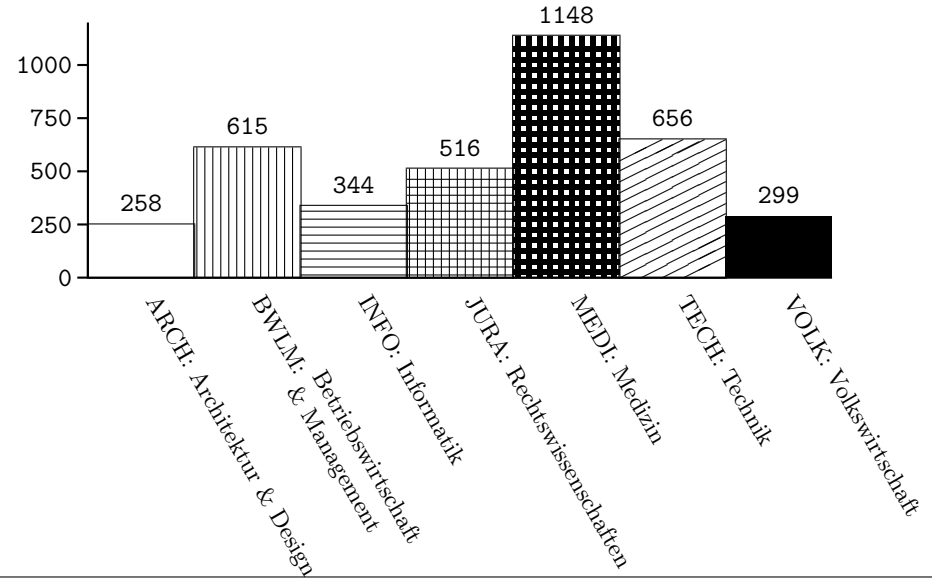


Figure 4.1: **Distribution of** SPRINGER **documents per category.**

4.1.2.2 Amazon Data Set

Name	AMAZON
Source	Descriptions of German books from Amazon.[9] Information was taken from content descriptions, reviews, publisher's announcements and author descriptions. For most titles only a part of these four were available. Book title and author were *not* included separately (but may occur in the running text). Reviewers' names were deleted as far as possible. The information being very different for different books, proper data cleansing proved very difficult and clustering may therefore have become a more challenging task.
Categories	Books were retrieved via the Amazon content categories. Some of the small categories were then merged, leading to the labels in Figure 4.2.
Number of Categories	21
Number of Documents	69,583
Documents per Category	323–15,412 (average: 3,313)
Document Length (Tokens)	20–3,248 (average 187)
Document Length (Types)	15–1,142 (average 126)
Tokens in Corpus	13,006,027
Types in Corpus	527,266
Token Length	2–202 letters (average: 5.98)

In many respects the AMAZON set was the most "difficult" data set. Encoding and presentation differed significantly between and sometimes even within documents. Among the consequences are the occurrence of a few irregular "tokens" (ca. 20) of lengths of up to 202 which do not have any ordinary meaning. Moreover, certain parts or even entire descriptions were found to be exact duplicates (with different ISBNs), and despite strenuous effort some near- or total duplicates survived the elimination process (as was found only much later).

In itself these difficulties are none too severe because they represent typical "real-world" phenomena that cluster applications will always face. On the other hand, higher-quality data is, of course, desirable as the results are less likely to be influenced by random effects.

[9] `www.amazon.de`

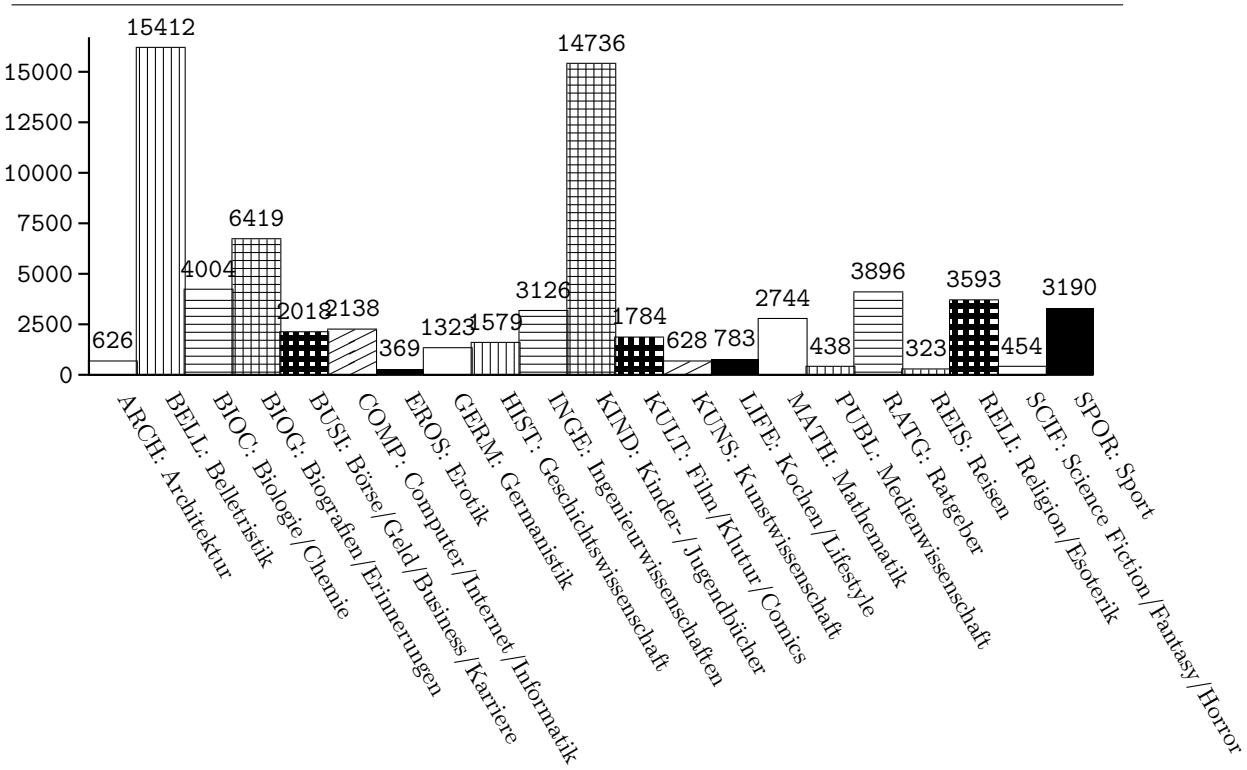


Figure 4.2: **Distribution of AMAZON documents per category.**

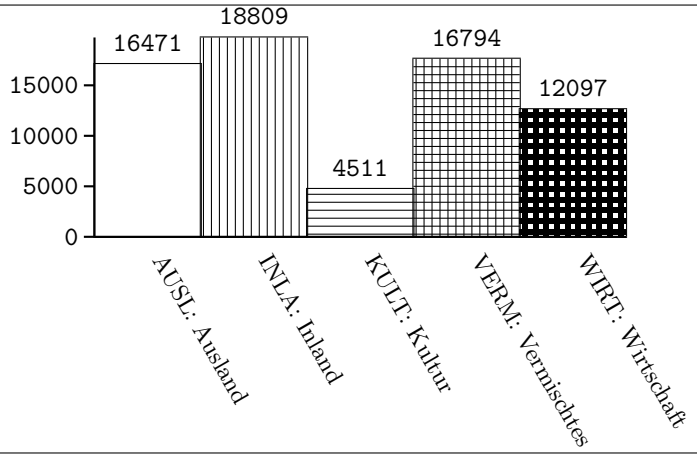


Figure 4.3: **Distribution of SDA documents per category.**

4.1.2.3 *Schweizerische Depeschenagentur* Data Set

Name	SDA
Source	Collection of newswire articles of the year 2004, courtesy of the *Schweizerische Depeschenagentur (SDA).*[10]
Categories	German SDA news come in five major categories. See Figure 4.3.
Number of Categories	5
Number of Documents	68,682
Documents per Category	4,511–18,809 (average: 13,736)
Document Length (Tokens)	20–992 (average: 192)
Document Length (Types)	18–524 (average: 133)
Tokens in Corpus	13,182,229
Types in Corpus	413,954
Token Length	2–50 letters (average: 6.22)

[10] `www.sda.ch`

4.1.2.4 Wikipedia Data Set

Name	WIKI
Source	Collection of articles from the German version of the free Internet encyclopaedia *Wikipedia*.[11]
Categories	Entries in the Wiki encyclopaedia are assigned to one or more categories. These are hierarchically organised. For our purposes most of the top categories were used. Documents which belonged to several main categories were assigned partly manually and partly through a simple majority algorithm to a single category. If the decision was too difficult, they were discarded. The categories are shown in Figure 4.4.
Number of Categories	22
Number of Documents	56,047
Documents per Category	321–6,483 (average: 2,548)
Document Length (Tokens)	20–18,534 (average: 279)
Document Length (Types)	15–4,076 (average: 167)
Tokens in Corpus	15,660,537
Types in Corpus	893,408
Token Length	2–202 letters (average: 6.33)

Data cleansing was slightly complicated by various HTML codes which were not all successfully eliminated (as demonstrated by the survival of half a dozen very long code-strings). Clustering results are hardly affected by these very rare phenomena. However, the fact that various documents had an overlapping nature and might have been assigned to more than one category may have made clustering more difficult.

[11] de.wikipedia.org

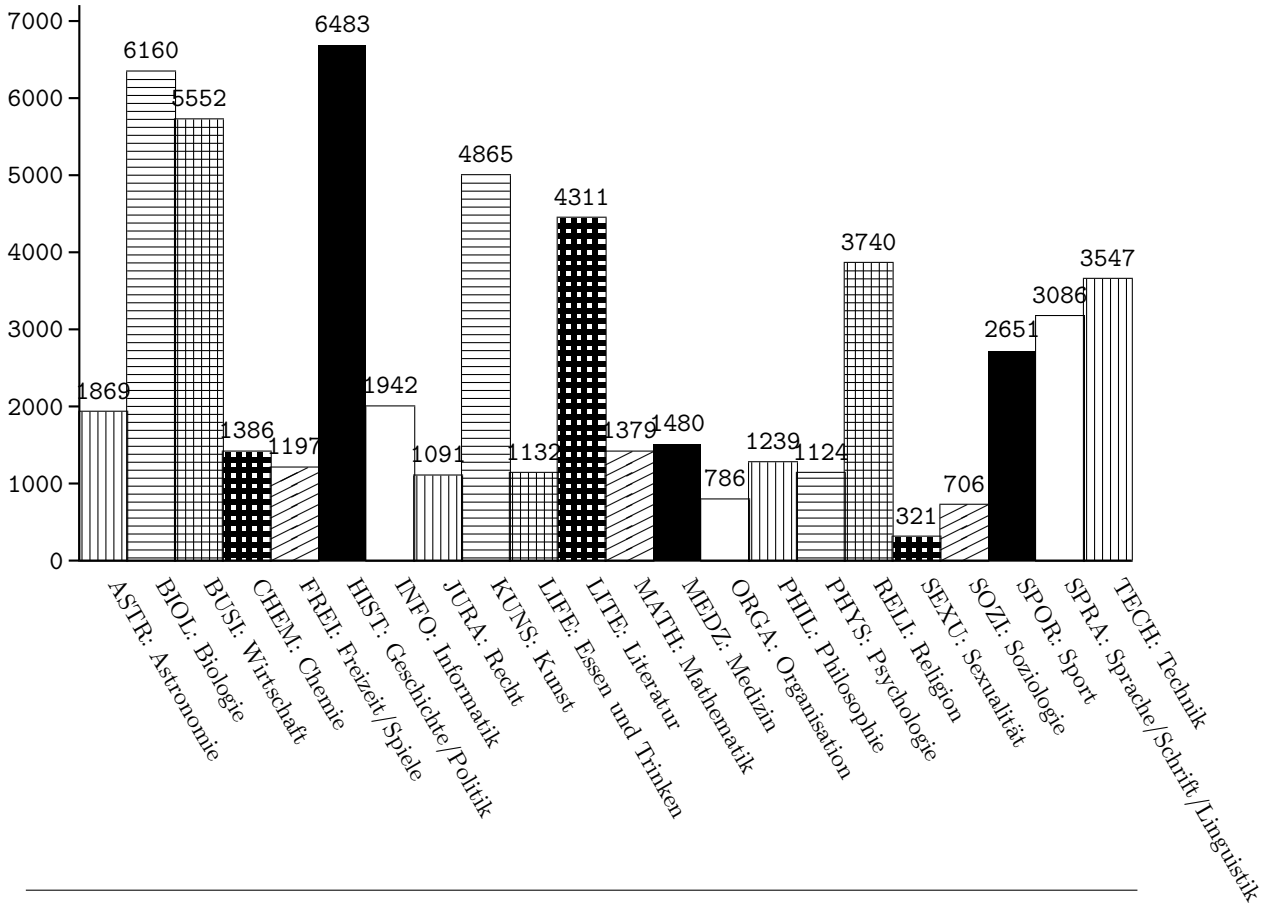


Figure 4.4: **Distribution of** WIKI **documents per category.**

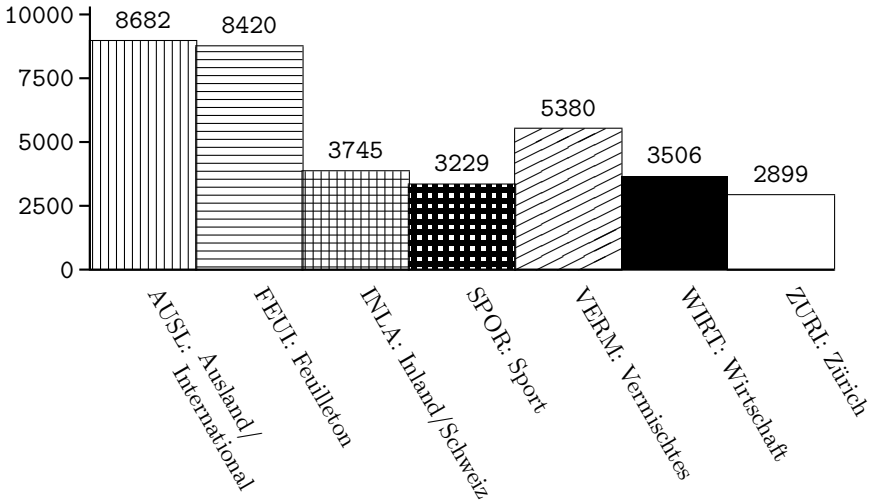


Figure 4.5: **Distribution of NZZ documents per category.**

4.1.2.5 *Neue Zürcher Zeitung* Data Set

Name	NZZ
Source	A selection of newspaper articles from 1993, courtesy of the *Neue Zürcher Zeitung (NZZ)*.[12] Author's names and short signatures, newswire abbreviations and date/place-lines as well as titles and subtitles have been omitted. Sport results, stock prices, content lists etc. were excluded.
Categories	Texts were taken from and labelled with the seven major departments of the paper. See Figure 4.5. The number of documents that was included per category was randomly chosen. Therefore the relative numbers do not reflect the actual frequencies of articles in the respective departments.
Number of Categories	7
Number of Documents	35,861
Documents per Category	2,899–8,682 (average: 5,123)
Document Length (Tokens)	342–3,505 (average: 726)
Document Length (Types)	139–1,691 (average: 422)
Tokens in Corpus	26,033,926
Types in Corpus	808,497
Token Length	2–69 letters (average: 6.25)

[12] www.nzz.ch

	SPRINGER	AMAZON	SDA	WIKI	NZZ
Categories	7	21	5	22	7
Documents	3,836	69,583	68,682	56,047	35,861
Avg. documents/category	548	3,313	13,736	2,548	5,123
Avg. length (in tokens)	88	187	192	279	726
Avg. length (in types)	70	126	133	167	422
Types in corpus	46,232	527,266	413,954	893,408	808,497
Tokens in corpus	335,658	13,006,027	13,182,229	15,660,537	26,033,926
Avg. token length	7.24	5.98	6.22	6.33	6.25

Table 4.3: **Summary of the five German data sets.**

4.1.2.6 Summary

The five experimental data sets are summarised in Table 4.3. Although in terms of content there are some similarities (two news corpora and two books corpora), the numbers show that no two sets are directly comparable and a large number of variations in terms of sizes, lengths and text diversity is covered.

Table 4.4 shows the distribution of part-of-speech categories for the five corpora (see Section 5.2.1 below for details and explanations). Figure 4.6 illustrates the distribution of the main word classes.

In view of the different text lengths, corpus sizes and scopes, direct *numeric* comparisons between the clustering results of two corpora require great caution and are likely to be invalid.

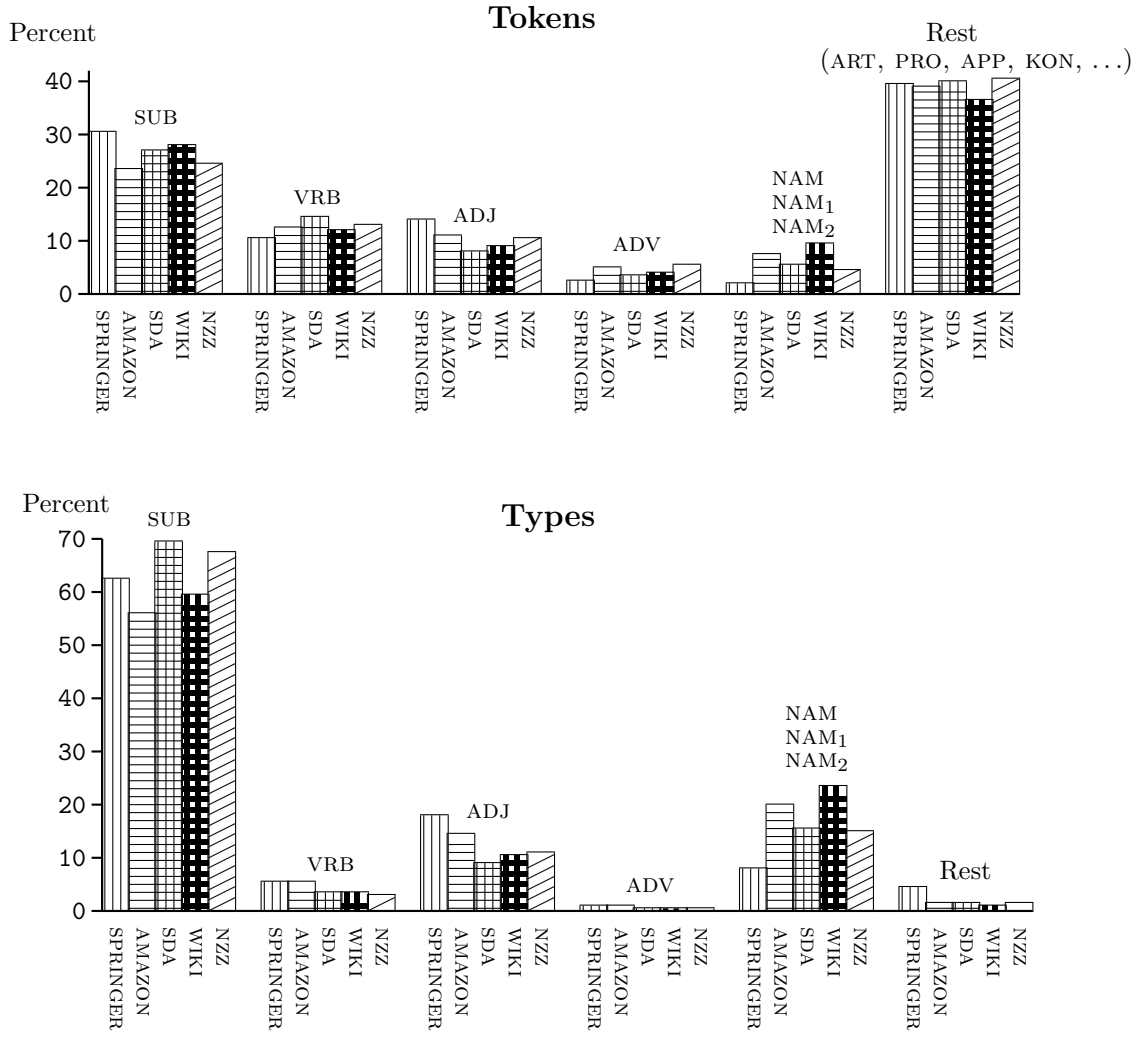


Figure 4.6: **POS distributions of tokens and types.** The bars reflect the respective proportions of the open word classes, with the closed classes being summed under "Rest" (see Table 4.4 for the exact figures). The individual POS labels are discussed in Section 5.2.1.

POS	SPRINGER		AMAZON		SDA		WIKI		NZZ	
	tokens	*types*	*tokens*	*types*	*tokens*	*types*	*tokens*	*types*	*tokens*	*types*
SUB	0.3073	0.6287	0.2393	0.5639	0.2740	0.6949	0.2805	0.5978	0.2486	0.6776
VRB	0.1085	0.0567	0.1281	0.0592	0.1487	0.0361	0.1195	0.0356	0.1326	0.0320
ADJ	0.1396	0.1797	0.1123	0.1454	0.0804	0.0908	0.0940	0.1094	0.1053	0.1125
ART	0.1383	0.0001	0.1040	0.0000	0.1417	0.0000	0.1175	0.0000	0.1331	0.0000
PRO	0.0552	0.0037	0.0945	0.0005	0.0570	0.0006	0.0632	0.0003	0.0760	0.0003
APP	0.1085	0.0023	0.0941	0.0004	0.1331	0.0004	0.1109	0.0002	0.1176	0.0003
KON	0.0716	0.0012	0.0595	0.0001	0.0404	0.0002	0.0506	0.0001	0.0533	0.0001
ADV	0.0270	0.0109	0.0541	0.0112	0.0383	0.0056	0.0420	0.0063	0.0569	0.0053
NAM	0.0158	0.0663	0.0444	0.1832	0.0428	0.1377	0.0643	0.2237	0.0312	0.1384
NAM_1	0.0024	0.0090	0.0212	0.0093	0.0089	0.0090	0.0199	0.0064	0.0082	0.0063
PTK	0.0117	0.0023	0.0187	0.0009	0.0174	0.0008	0.0139	0.0005	0.0197	0.0006
UNK	0.0089	0.0293	0.0149	0.0144	0.0021	0.0125	0.0054	0.0098	0.0028	0.0172
NAM_2	0.0016	0.0077	0.0099	0.0103	0.0064	0.0104	0.0109	0.0072	0.0079	0.0080
NUM	0.0035	0.0019	0.0049	0.0012	0.0088	0.0012	0.0074	0.0025	0.0068	0.0013

Table 4.4: **POS distributions.** Relative distribution of the part-of-speech tags within the five data sets.

4.2 Software

All experiments in this study have been performed with the fast clustering toolkit CLUTO (Karypis, 2003). CLUTO is a set of powerful general-purpose clustering algorithms which have, at least partly, been designed for documents (Zhao and Karypis, 2001, 2002, 2003). The software is freely available[13] and has immediately gained a large following, in particular in the IR area (e. g. Casillas *et al.*, 2003; Boulis and Ostendorf, 2004; Purandare and Pedersen, 2004; Zhong and Ghosh, 2005).

CLUTO includes several clustering algorithms (graph-based, partitional, hierarchical agglomerative and divisive) which can be run with a variety of criterion functions. As input it accepts either a document-document similarity matrix or a document-feature matrix. Several parameters can be used to control the clustering procedure. We relied mostly on the default options. See Section 4.4 below for further details.

4.3 Evaluation Methodology

Our primary focus is on the *quality* of a clustering solution (Section 4.3.1). However, in a number of situations a method can in addition be evaluated from a *quantitative* aspect (Section 4.3.3). Finally, the problem of interpreting multiple results is briefly discussed (Section 4.3.4).

4.3.1 Cluster Validity

Using labelled data enabled us to use an external and relatively objective evaluation criterion to measure the relative success of different document representation and clustering techniques. Tables 4.5 and 4.6 show the average correlation of different evaluation measures from Section 2.6.2. For the first table of the two, 4,204 ordinary cluster solutions from subsequent experiments were compared, while 145 random cluster assignments were examined for the second table.

Most measures appear to be relatively closely correlated, with the biggest exceptions being the Rand coefficient for well-behaved clustered data and the Γ statistic for randomly distributed data.

The $\mathcal{Q}_0$ measure propagated in Dom's comparative study (2002) is the theoretically best-supported measure. If the number of clusters equals the number of labels (as in our case), it is equivalent to weighted entropy (Eq. 2.76).[14] In our further experiments we thus used weighted entropy values which are one of the two values already provided by CLUTO (the other is weighted purity).

In the following chapters each document representation method is thus evaluated by the weighted entropy of the cluster solutions it produces. The smaller the entropy, the better the fit between clustering and original labelling and the better the representation method. In order to obtain more reliable results, each cluster run was performed not just once, but on ten random permutations of the ordered input set.[15]

[13] `www-users.cs.umn.edu/~karypis/cluto`

[14] This equivalence between weighted Entropy and $\mathcal{Q}_0$ is confirmed by the correlation coefficient of 0.999 in both tables.

[15] The ten permutations were the same for each experiment. The permutations became necessary after experiments with CLUTO's random initialisation parameter *–seed* had failed to produce the promised effect.

	Rnd	Jac	F/M	Γ	$\mathcal{Q}_0$	wP	uP	wMR	uMR	wE	uE	wF	uF
Rnd	1.000	0.008	0.016	0.221	0.123	0.100	0.139	0.033	0.072	0.104	0.273	0.018	0.026
Jac	0.008	1.000	0.996	0.972	0.928	0.937	0.827	0.935	0.956	0.938	0.851	0.978	0.949
F/M	0.016	0.996	1.000	0.971	0.941	0.946	0.848	0.948	0.962	0.951	0.872	0.987	0.957
Γ	0.221	0.972	0.971	1.000	0.881	0.942	0.791	0.920	0.951	0.895	0.779	0.964	0.932
$\mathcal{Q}_0$	0.123	0.928	0.941	0.881	1.000	0.960	0.912	0.869	0.932	0.999	0.949	0.945	0.965
wP	0.100	0.937	0.946	0.942	0.960	1.000	0.905	0.897	0.959	0.965	0.885	0.964	0.971
uP	0.139	0.827	0.848	0.791	0.912	0.905	1.000	0.844	0.859	0.904	0.970	0.877	0.903
wMR	0.033	0.935	0.948	0.920	0.869	0.897	0.844	1.000	0.950	0.880	0.845	0.965	0.910
uMR	0.072	0.956	0.962	0.951	0.932	0.959	0.859	0.950	1.000	0.942	0.863	0.981	0.983
wE	0.104	0.938	0.951	0.895	0.999	0.965	0.904	0.880	0.942	1.000	0.941	0.954	0.969
uE	0.273	0.851	0.872	0.779	0.949	0.885	0.970	0.845	0.863	0.941	1.000	0.886	0.911
wF	0.018	0.978	0.987	0.964	0.945	0.964	0.877	0.965	0.981	0.954	0.886	1.000	0.976
uF	0.026	0.949	0.957	0.932	0.965	0.971	0.903	0.910	0.983	0.969	0.911	0.976	1.000

Table 4.5: **Correlation of evaluation measures (with ordinary clusters).** Absolute correlation coefficients of different evaluation indices with *clustered* data. (Abbreviations: Rnd=Rand, Jac=Jaccard, F/M=Fowlkes/Mallows, Γ=Γ Statistic, $\mathcal{Q}_0$=Dom's $\mathcal{Q}_0$ measure, P=Purity, MR=Macro-Recall, E=Entropy, F=F-Measure, w=weighted, u=unweighted. See Section 2.6.2 for the definitions.)

	Rnd	Jac	F/M	Γ	$\mathcal{Q}_0$	wP	uP	wMR	uMR	wE	uE	wF	uF
Rnd	1.000	0.994	0.997	0.035	0.998	0.873	0.860	0.987	0.971	0.998	0.995	0.992	0.982
Jac	0.994	1.000	0.999	0.020	0.989	0.823	0.812	0.996	0.988	0.987	0.986	0.988	0.989
F/M	0.997	0.999	1.000	0.022	0.993	0.839	0.828	0.994	0.984	0.991	0.990	0.990	0.988
Γ	0.035	0.020	0.022	1.000	0.047	0.058	0.082	0.009	0.049	0.050	0.053	0.026	0.068
$\mathcal{Q}_0$	0.998	0.989	0.993	0.047	1.000	0.877	0.865	0.979	0.966	0.999	0.997	0.987	0.979
wP	0.873	0.823	0.839	0.058	0.877	1.000	0.987	0.797	0.743	0.889	0.882	0.871	0.808
uP	0.860	0.812	0.828	0.082	0.865	0.987	1.000	0.786	0.734	0.877	0.881	0.862	0.800
wMR	0.987	0.996	0.994	0.009	0.979	0.797	0.786	1.000	0.990	0.978	0.977	0.988	0.990
uMR	0.971	0.988	0.984	0.049	0.966	0.743	0.734	0.990	1.000	0.963	0.963	0.968	0.992
wE	0.998	0.987	0.991	0.050	0.999	0.889	0.877	0.978	0.963	1.000	0.998	0.990	0.980
uE	0.995	0.986	0.990	0.053	0.997	0.882	0.881	0.977	0.963	0.998	1.000	0.990	0.980
wF	0.992	0.988	0.990	0.026	0.987	0.871	0.862	0.988	0.968	0.990	0.990	1.000	0.987
uF	0.982	0.989	0.988	0.068	0.979	0.808	0.800	0.990	0.992	0.980	0.980	0.987	1.000

Table 4.6: **Correlation of evaluation measures (with random data).** Absolute correlation coefficients of different cluster evaluation indices with *random* clusters. (For the abbreviations see Table 4.5.)

Results are usually arranged in tabular form as follows:

	SPRINGER	AMAZON	SDA	WIKI	NZZ
method1	0.505 [0.005]	0.517 [0.003]	0.518 [0.002]	0.473 [0.008]	0.436 [0.011]
method2	0.523 [0.008]	0.488 [0.005]	0.530 [0.008]	0.436 [0.005]	0.451 [0.001]
method3	0.546 [0.014]	0.494 [0.002]	0.522 [0.001]	0.425 [0.003]	0.390 [0.001]

The first column indicates the specific algorithms, methods and parameters that were tested, while the following five columns show the average entropy values of the five test collections, with standard deviations in square brackets. Occasionally, technical reasons prevented some documents from being clustered (typically when a radical feature representation method left certain documents without any features at all). These results are either marked with an apostrophe or else the number of "lost" objects is indicated in a footnote.

Note: Meaningful numeric comparisons are only possible for different methods *within the same* data set. Numeric comparisons *across* the five data sets do not make sense.

4.3.2 Confusion Matrix and ROC Diagram

A popular means of visualising an individual cluster solution with regard to external labels is the *confusion matrix* which shows for each label-cluster pair its sum of common documents (i. e. the contingency table $\mathcal{H}$ from Section 2.6.2). For example:

	L_1	L_2	L_3	L_4	$\sum$
C_1	2	0	15	2	19
C_2	32	2	5	12	51
C_3	1	18	1	0	20
C_4	4	3	1	16	24
$\sum$	39	23	22	30	114

However, with an increasing number of dimensions the table soon becomes unwieldy. It is thus only very rarely used for illustration purposes.

Using the contingency table coefficients a_{00} to a_{11} defined in Equations 2.56 to 2.59, the clustering solution can also be characterised by its *recall* value ($\frac{a_{00}}{a_{00}+a_{01}}$) and its *precision* value ($\frac{a_{00}}{a_{00}+a_{10}}$). Alternatively, the clustering solution could be shown as a point in a ROC diagram (*Receiver Operating Characteristic* diagram), which plots *recall* on the x-axis versus *fallout* on the y-axis, with fallout being defined as $\frac{a_{10}}{a_{10}+a_{11}}$. Several clustering solutions can then be displayed in one comparative diagram, though the need to label the data points makes it here an unpractical tool if a larger number of cluster solutions under different parameter settings are to be compared.

Therefore, we relayed on the tabular form mentioned at the end of the previous section. In fact, in most cases we derived simple visualisations (cf. the next section for an example) while the underlying exact numbers are listed in Appendix D.

4.3.3 Matrix Size

The *quantitative* impact of different document representation techniques on clustering can be measured in three ways:

- *by time:* the time that is necessary to calculate a cluster solution from the input matrix.

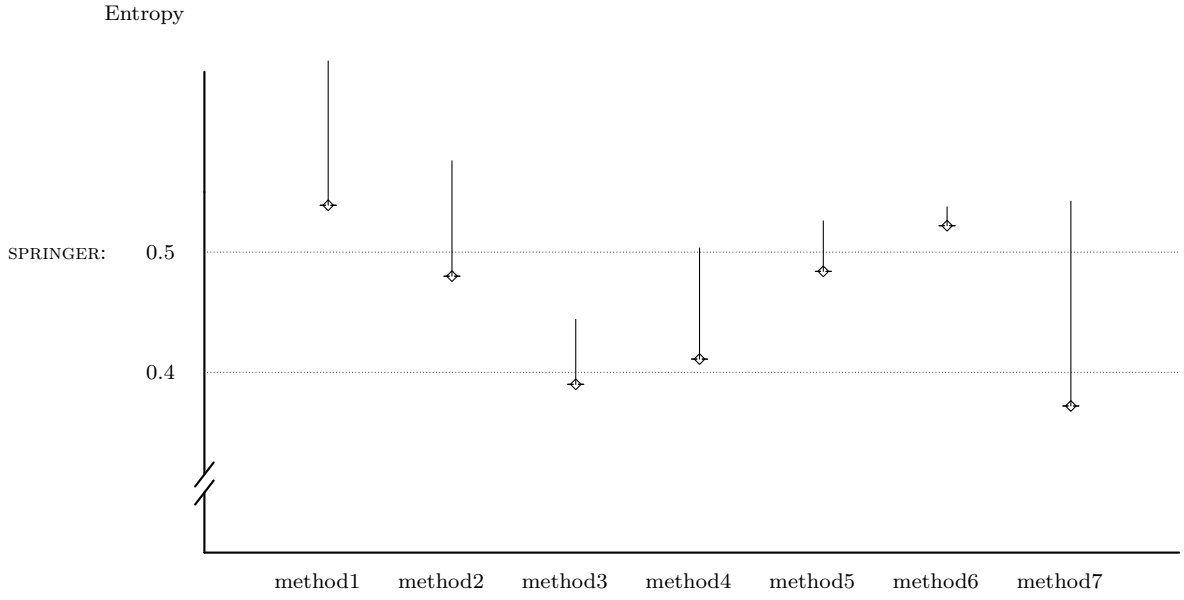
- *by dimensionality:* the number of features (columns) of the input matrix.
- *by matrix size:* the number of non-zero elements in the input matrix.

Since time requirements are often difficult to measure exactly and strongly dependent on the hard- and software, a matrix-dependent measure was preferred. With sparse and high-dimensional matrices (such as those in document clustering) there can be significant differences in the way dimensionality and matrix size vary. For most of CLUTO's cluster algorithms the time and memory requirements were observed to be more closely related to the number of non-zero elements than to the number of matrix dimensions, so **matrix size** was selected to measure the quantitative aspects.

The matrix size is usually indicated by percentages in round parentheses after the entropy values. The percentage values show the relation between matrix size before and after application of the given method:

	SPRINGER	AMAZON	SDA	WIKI	NZZ
base	0.425 (100%)	0.455 (100%)	0.449 (100%)	0.381 (100%)	0.351 (100%)
reduction method 1	0.412 (67%)	0.460 (71%)	0.446 (70%)	0.393 (77%)	0.348 (80%)
reduction method 2	0.448 (70%)	0.460 (75%)	0.447 (71%)	0.394 (79%)	0.350 (82%)
reduction method 3	0.406 (68%)	0.454 (70%)	0.438 (70%)	0.389 (79%)	0.350 (80%)

We visualise this secondary target variable (the percentages) by vertical lines on top of the entropy data points, as in this diagram:



The length of each line is directly related to the percentage number. Thus, the shorter the line and the further near the bottom of the figure (= the lower the entropy), the better

is the corresponding clustering method. From the example, we could conclude that method 7 was performing best but required an extraordinarily large amount of computational resources. Method 3 might therefore be preferable as it provides a good result with a much smaller matrix size.

In order to numerically evaluate a series of experiments in which both the qualitative and the quantitative aspects are of interest, the two measures can be roughly combined by assigning ascending rank scores to each entropy value and also to each matrix size value (for each of the five data sets separately). Thus, each experimental run receives two rank scores which can be added to yield a rough measure of comparison *within the given series of experiments*.

For instance, for the WIKI data set in the previous example the rank scores would be as follows:

	Entropy:	value	rank	**Size:**	value	rank	**Rank-sum**
reduction method 1		0.393	2		77%	1	3
reduction method 2		0.394	3		79%	2.5	5.5
reduction method 3		0.389	1		79%	2.5	3.5

Reduction method 1 would thus have the lowest rank-sum and be considered best. Being a non-parametric comparison method, rank-sum must be interpreted with appropriate care. On the other hand, unlike the entropy or matrix size values the rank-sums can also be added up across all five data sets.

4.3.4 Interpreting Multiple Experiments

When working with several data sets and a large number of different scenarios (algorithms, parameters, etc.), the question naturally arises how to interpret the results. IR is primarily concerned with practical systems and the motto "it's good as long as it works" is often adhered to. Nevertheless, statistical significance tests to back up research results would be welcome. See Hull (1993) for a discussion of statistical evaluation methods in IR as well as a few arguments pro and contra their adoption.

Generally speaking, an Analysis of Variance (ANOVA) for Repeated Measures would be an appropriate evaluation tool for our experiments. However, there are only five data sets and, moreover, it appears very questionable whether the clustering results follow a normal distribution. For these two reasons the use of an ANOVA cannot be justified after all.

For non-parametric alternatives (for instance the Friedman test) the paucity of data sets is even graver. It then becomes virtually impossible to prove an effect at a 5% significance level. With no suitable statistical method thus available, the interpretation of the results had therefore, albeit reluctantly, to be conducted in an "intuitive" fashion. Nevertheless, the large number of individual experiments reported as well as the indication of standard deviations should still allow the careful reader to form an opinion and put the results into a perspective.[16]

4.4 Algorithms and Parameters

CLUTO offers a wide choice of clustering algorithms and settings. This section describes our main choices as well as a few preliminary experiments motivating these decisions. Table 4.7 shows the actual parameter choices for the experiments in the following chapters. A brief discussion follows in the sub-sections.

[16] With regard to the standard deviation it should be kept in mind, however, that the ten clustering runs of each experiment were quite often *not* normally distributed.

Parameter	*Choice*	*Alternatives*	*Comment*
(program)	**vcluster**	scluster	All experiments were conducted with the *vcluster* program which takes a vector matrix as input, whereas *scluster* works with similarity matrices.
–clmethod	**rbr**	rb, direct, agglo, graph, bagglo	For explanations of the different algorithms and a comparative experiment see Section 4.4.1.
–sim	**cos**	corr, [dist], [jacc]	Cosine similarity (the default option) was given preference to the correlation coefficient. Jaccard and Euclidean distance are only applicable with the graph algorithm.
–crfun	**i2**	i1, e1, g1, g1p, h1, h2, slink, wslink, clink, wclink, upgma	As criterion function the default option $\mathcal{I}_2$ was selected. The alternatives refer to the different criterion functions discussed in Section 2.3.1.1 resp. 2.4.1, whereby *slink, wslink, clink, wclink* refer to weighted and unweighted formulae for single- and complete-link.
–cstype	**largess**	large, best	The "large sub-size" criterion for sparse and high-dimensional matrices was selected to determine which cluster to bi-sect next. For experiments with the alternatives see Section 4.4.1.
–rowmodel	NONE	MAXTF, SQRT, LOG	Local waiting was turned off (default option). For experiments with local waiting see Section 4.4.2.
–colmodel	IDF	NONE	Global weighting with inverse document frequency $g(D_{\cdot j}) = \log_2 \frac{n}{n_j}$ was applied. In reality, the model was further extended to a squared variant of IDF—see the experiments in Section 4.4.2.
–ntrials	**10**	$\mathbf{N}^+$	Ten cluster solutions were computed for each input matrix and the one performing best with regard to the internal cluster criterion was selected (CLUTO's default choice).
–niter	**10**	$\mathbf{N}^+$	A maximum of 10 refinement steps was performed at each clustering step (CLUTO's default choice).

Table 4.7: **Parameter choices for Cluto** (cf. Karypis, 2003).

4.4.1 Choice of Algorithm

The choice of the *repeated bi-sectional algorithm with refinement (rbr)* was based on preliminary experiments with a bag-of-lemmata model after stopword removal.[17] The results of the experiment are shown in Table 4.8 (page 100) and summarised below. Figure 4.7 (page 101) illustrates the same results graphically.

agglo: The *agglomerative* clustering method (cf. Section 2.4.1) turned out to be too demanding on the memory resources for all but the SPRINGER data set. Its result with the latter does not compare well with the other algorithms.

bagglo: The *biased agglomerative* algorithm (agglomerative with a partitional pre-processing step, cf. Section 2.4.1.5) encountered the same restrictions as the pure agglomerative method. For the SPRINGER set it showed a clearly better result than the former.

direct: The *direct*, simultaneous partitional method (a k-means algorithm, cf. Section 2.3.1) worked quite well for all data sets except the SPRINGER set.

graph: The *graph-partitional* method (using the MINcut criterion, cf. Section 2.1.3) showed excellent results for the SPRINGER set—far better than all other methods. However, for the other four (larger) sets the nearest-neighbour graph was disconnected, leading to additional clusters and various documents being not clustered at all. For the purpose of reliably comparing different representation techniques this clustering method was therefore not suitable.

rb: The *repeated bi-sectional* method (cf. Section 2.4.2) worked generally well. Of the three *–cstype* parameter choices (governing which cluster to split next) "large" and "largeSS" were better than "best". "Large" selects the largest cluster, "best" the cluster whose bi-section optimises the cluster criterion function, "largeSS" the cluster whose bi-section leads to the largest reduction in dimensions accounting for the majority of within-cluster similarity of the objects.

rbr: The *repeated bi-sectional refined* method (cf. Section 2.4.2) was overall the most successful and selected for the further experiments. For *–cstype* the value "largeSS" was chosen which the CLUTO manual recommends for sparse and high-dimensional data sets as are typical of document clustering.

Table 4.9 gives a brief indication of the (relative) time demands of the different algorithms. Clustering was performed with the SDA and SPRINGER data sets on a SunBlade 1500 with a 1.1 GHz UltraSPARC-IIIi processor and 1GB RAM.

4.4.2 Choice of Weighting Scheme

The general basics of feature weighting were described in Section 3.3, where the general weighting model $d'_{ij} = t(d_{ij}) \cdot g(d_{.j}) \cdot s(d_{i.})$ was introduced.

Since the cosine has been chosen as the similarity measure, (Euclidean) *normalisation* $s(d_{i.})$ becomes irrelevant. Thus, only *local* and *global* weighting components had to be evaluated.

[17] The actual details of the bag-of-lemmata model and the stopword removal procedure are further discussed in the next chapter.

	SPRINGER	AMAZON	SDA	WIKI	NZZ
agglo	0.857	n/a	n/a	n/a	n/a
bagglo	0.540	n/a	n/a	n/a	n/a
direct	0.539 [0.013]	0.506 [0.010]	0.518 [0.003]	0.426 [0.009]	0.430 [0.029]
graph	0.430 [0.003]	0.562[a] [0.005]	0.566[b] [0.007]	0.518[c] [0.004]	0.622[d] [0.020]
rb (best)	0.526 [0.007]	0.507 [0.002]	0.571 [0.006]	0.479 [0.005]	0.467 [0.001]
rb (large)	0.548 [0.014]	0.506 [0.002]	0.518 [0.002]	0.458 [0.005]	0.429 [0.003]
rb (largess)	0.505 [0.005]	0.517 [0.003]	0.518 [0.002]	0.473 [0.008]	0.436 [0.011]
rbr (best)	0.523 [0.008]	0.488 [0.005]	0.530 [0.008]	0.436 [0.005]	0.451 [0.001]
rbr (large)	0.546 [0.014]	0.494 [0.002]	0.522 [0.001]	0.425 [0.003]	0.390 [0.001]
rbr (largess)	0.491 [0.008]	0.496 [0.002]	0.522 [0.001]	0.410 [0.008]	0.402 [0.022]

[a] 2 extra clusters, 300 documents not clustered.
[b] 5 extra clusters, 310 documents not clustered.
[c] 3 extra clusters, 455 documents not clustered.
[d] 4 extra clusters, 236 documents not clustered.

Table 4.8: **Comparison of Cluto's different clustering algorithms** (–**clmethod** parameter); in parentheses (best, large, largess) different values for the –**cstype** parameter. For explanations of the abbreviations see the text. (The table shows entropy values and in square brackets standard deviations.)

	SPRINGER	SDA
agglo	22.0s	n/a
bagglo	24.2s	n/a
direct	2.6s	132s
graph	5.1s	870s
rb (best)	2.1s	122s
rb (large)	1.8s	104s
rb (largess)	2.2s	121s
rbr (best)	2.3s	133s
rbr (large)	1.9s	115s
rbr (largess)	2.5s	132s

Table 4.9: **Time demands of different clustering algorithms** for two data sets. The agglomerative algorithms could not cope with the SDA data set because of memory overflow.

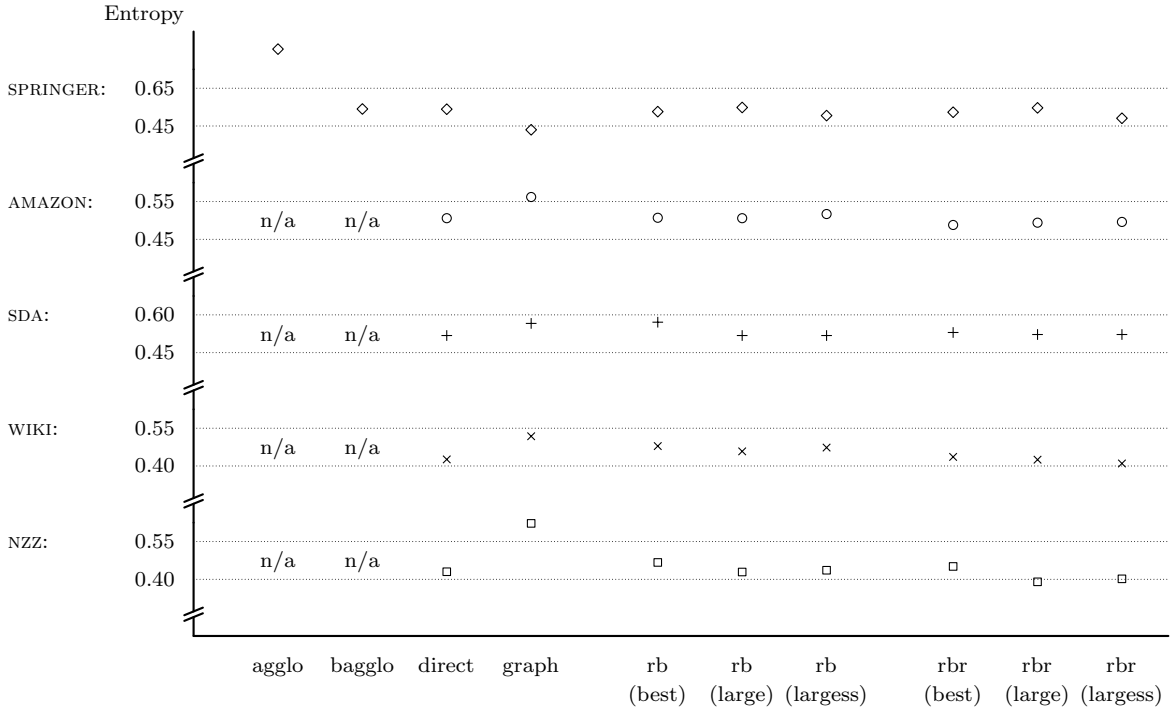


Figure 4.7: **Visual comparison of CLUTO's different clustering algorithms.** For the underlying numbers see Table 4.8. Lower values indicate better clustering results. For clarity's sake the five data series are shown in five separated sections of the diagram.

CLUTO offers these built-in weighting options:

Local weighting (*–rowmodel* parameter):

	$t(d_{ij}) =$
NONE:	d_{ij} (normal frequency),
MAXTF:	$0.5 + 0.5\frac{d_{ij}}{\max_l d_{il}}$,
LOG:[18]	$\text{sgn}(d_{ij}) * \log_2 \lvert d_{ij} \rvert$,
SQRT:	$\text{sgn}(d_{ij}) * \sqrt{\lvert d_{ij} \rvert}$.

Global weighting (*–colmodel* parameter):

	$g(d_{.j}) =$
NONE:	1,
IDF:	$\log_2 \frac{n}{\text{df}(j,H)}$.[19]

Figure 4.8 shows the results of all possible combinations of these weighting models. The same input data was used as in the previous section.

As expected, smoothing term frequencies with IDF improved performance dramatically. The overall effect of local frequency adjustments was less clear-cut, though it is evident that the two news corpora did profit from local frequency smoothing.

4.4.2.1 Inverse Document Frequency Squared

More or less by accident, we then extended the scheme by applying a two-step weighting scheme:

The "Double-Weighting Scheme".

1. An "external" LOG-IDF weighting was applied on all matrix values, using the *natural logarithm*[20] with $t(d_{ij}) = 1 + \ln d_{ij}$ and $g(d_{.j}) = \ln(n/\text{df}(j,H))$.
2. Then an additional weighting step was performed with CLUTO's built-in row and column model parameters (of which the NAME-IDF combination was favoured and used as default in the further chapters).

The effects of this *double-weighting scheme* are shown in Figure 4.9. The results appear to be very promising: double-weighting led to improvements throughout our experiments. First, it may be noted that—accidentally or not—the binary logarithm performed somewhat worse than the natural logarithm for all data sets (LOG-IDF in the first versus NONE-NONE in the second table).

[18] Should only be used for non-negative numbers as it is rarely desirable to map a number and its negative inverse on one and the same value (e. g. $d_{ij} = 4$ and $d_{ij} = -\frac{1}{4}$ would lead to the same result)

[19] It was actually impossible to reproduce this effect externally, i. e. weighting the vectors with $\log_2(n/\text{df}(j,H))$ before feeding them into CLUTO led to different results than applying the IDF-parameter, and the differences seemed too big to be fully accounted for by mere rounding differences.

[20] I. e. the more common logarithm with base e (*natural logarithm*)was used for the external weighting, whereas CLUTO works with the *binary logarithm* (base 2).

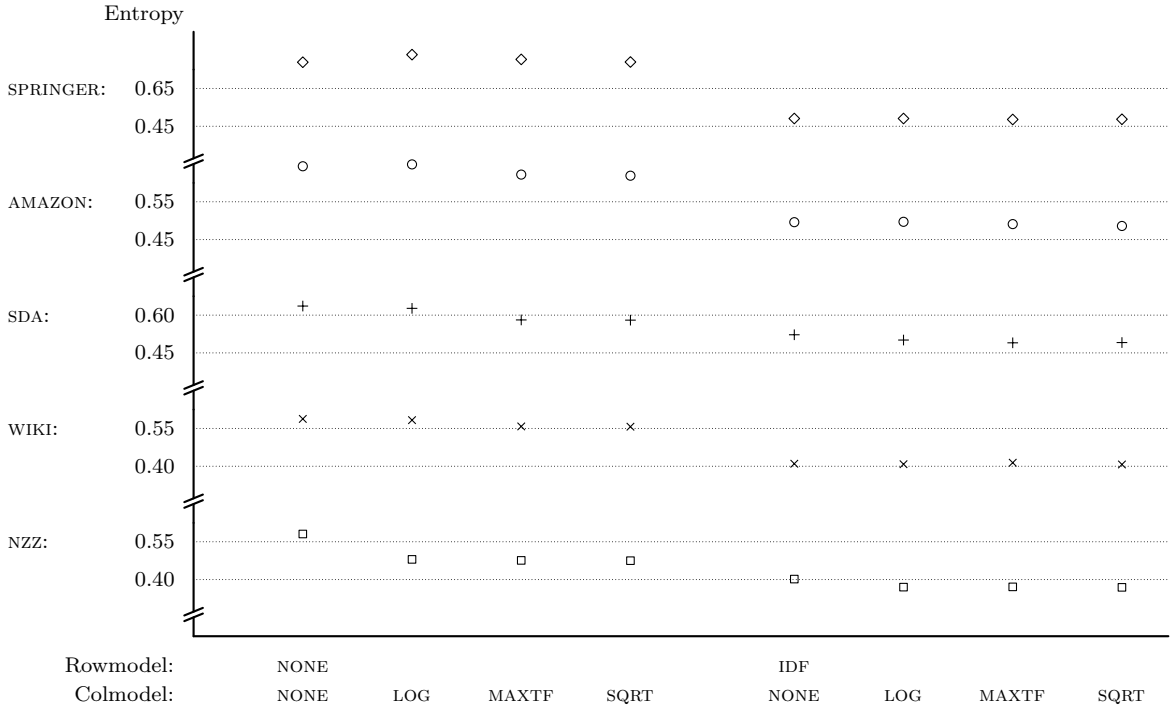


Figure 4.8: **Evaluation of Cluto weighting models**, with the *rbr(largess)* algorithm and different options for the *–colmodel* and *–rowmodel* parameters. (For the numbers underlying this figure refer to Table D.1.)

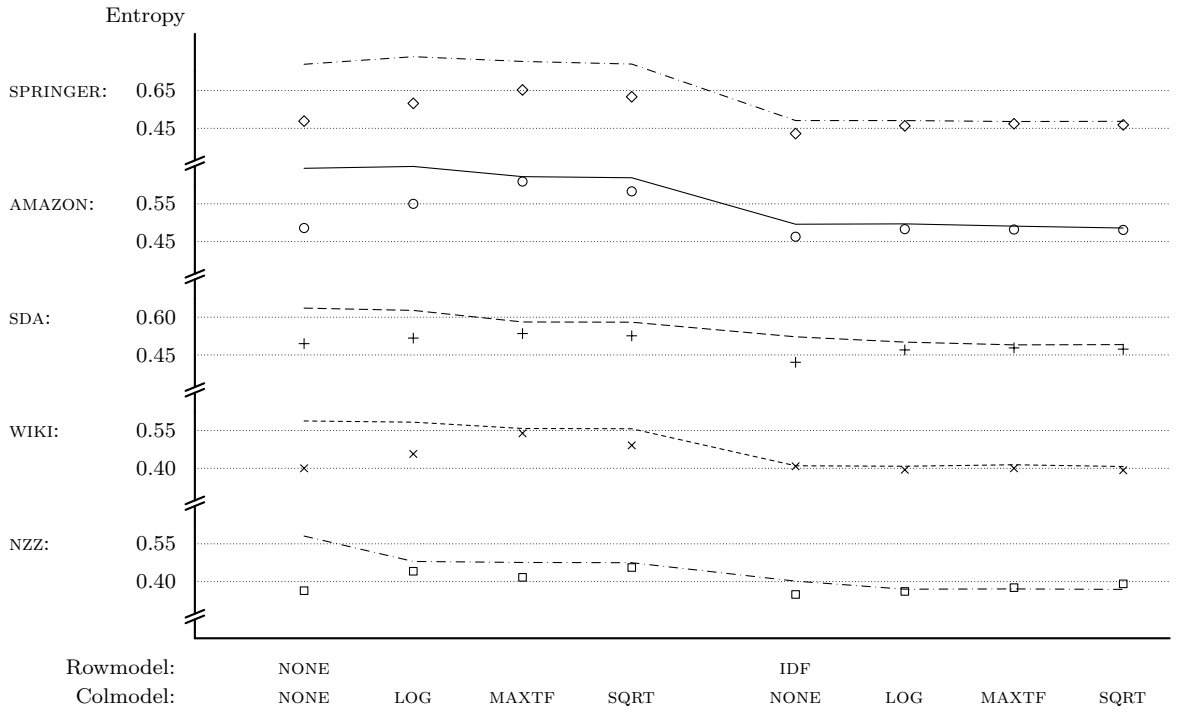


Figure 4.9: **Double-weighting**. Evaluation of CLUTO's built-in weighting models *after* prior application of an external LOG-IDF weighting scheme to the data. The connected lines repeat the results from Figure 4.8 and show that the new results are almost always better. (For the underlying numbers refer to Table D.2.)

More important, however, is the fact that twofold application of IDF seems to improve performance in four of five cases. The best combination was external LOG-IDF together with CLUTO's NONE-IDF weighting. This weighting scheme—effectively resulting in

$$\begin{aligned} {d'}_{ij} &= \phi_{\text{cluto}}(\phi_{\text{external}}(d_{ij}, H), H) \\ &= t_{\text{external}}(d_{ij}) \cdot g_{\text{external}}(d_{\cdot j}) \cdot g_{\text{cluto}}(d_{\cdot j}) \\ &= (1 + \ln d_{ij}) \cdot \ln \frac{n}{\text{df}(j, H)} \cdot \log_2 \frac{n}{\text{df}(j, H)} \end{aligned} \tag{4.1}$$

—was therefore used throughout the following chapters. Repeated comparisons at later stages of our work confirmed this choice, as the results were virtually always much better than when relying on CLUTO's internal weighting options alone.

The double-weighting scheme of Equation 4.1 will be further referred to as LOG-IDF^2 and the global component alone ($g_{\text{external}} \cdot g_{\text{cluto}}$) as "IDF squared".

4.4.2.2 Validity of IDF Squared

Heretofore, the squared variant of IDF weighting has been very rarely used in IR (two recent exceptions being Goharian *et al.* 2001 and Henzinger *et al.* 2003). To our knowledge it has not yet been tested for clustering at all. As we found that throughout our study the LOG-IDF^2 scheme out-performed the standard alternatives, we decided to perform a few more tests with IDF^2 in order to establish whether the effect may not have been purely accidental or caused by a quirk of CLUTO's term weighting component.

Figure 4.10 reports on the results with a number of different external/internal combinations, as well as natural/binary logarithms. Columns 2–7 report on experiments with simple IDF, while columns 8–14 report on experiments with IDF^2. It transpires that neither the choice of logarithm nor the distribution between external and internal weighting makes a difference. The effect of IDF^2 versus IDF is stable: four data sets with a difference in favour of IDF^2, and only one case with inconclusive data (WIKI).

In addition, it can again be observed that for the NZZ set more than any other it is crucial to smooth term frequencies locally. One reason could be the higher average text length of the NZZ corpus, leading to more skewed within-document frequency distributions.

In a second attempt to validate the usability of IDF^2 we applied the same measures to the three English test document sets that form part of the CLUTO standard distribution. With sizes of 204 to 8,580 documents, they are distinctively smaller than the average of our German data sets (although on average the texts are longer). Table D.4 reports on the results of this small comparative study. They support IDF^2 partially. First, we note that logarithmic local weighting, which has been found generally useful for our German data sets (at least not harmful), tended to worsen the scores of the CLUTO standard data sets. Comparing the bare IDF and IDF^2 scores (columns 2–4 and 8–9), we find each method favoured in one case, while in the case of the *tr23* set, there was no real difference.

Conclusions. The above experiments showed that for clustering the *inverse document frequency squared* (IDF^2) must be considered a serious alternative to common IDF. Of the eight data sets examined, five showed better results with IDF^2, two barely any differences and only one set (CLUTO's "sports" set) fared better with simple IDF. Thus, IDF^2 deserves to be tested more extensively also in future work.

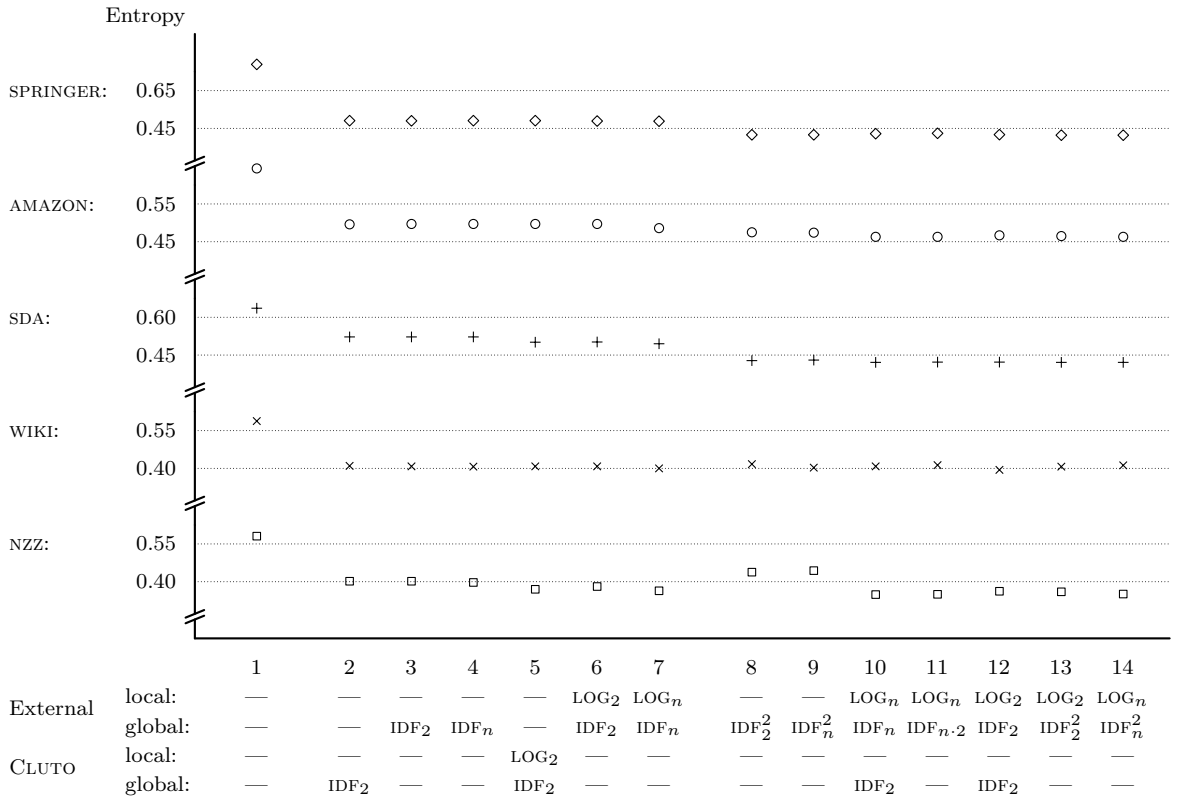


Figure 4.10: **Evaluation of different** IDF **and** IDF^2 **variants.** IDF$_n$ and LOG$_n$ refer to formulae using the natural logarithm, IDF$_2$ and LOG$_2$ to those with the binary logarithm, LOG$_{n\cdot 2}$ refers to the combined use of IDF$_n$ followed by IDF$_2$, IDF$_2^2$ and IDF$_n^2$ to the twofold application of IDF$_2$ and IDF$_n$. (For the underlying numbers refer to Table D.3.)

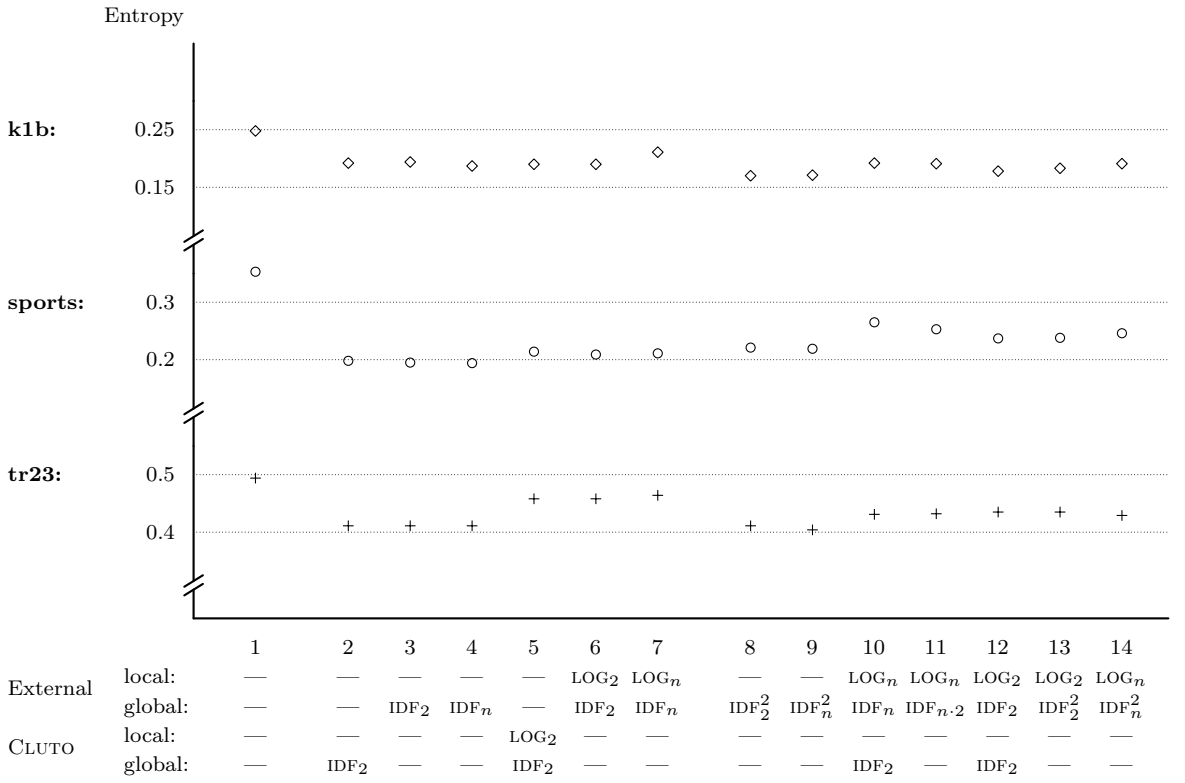


Figure 4.11: **Experiments with three standard sets.** Different weighting schemes applied to the three test collections *k1b*, *sports* and *tr23* coming as part of the CLUTO standard distribution. Scores on the left refer to normal weighting, scores on the right to IDF^2 weightings. (For the underlying numbers refer to Table D.4.)

Chapter 5

Reduced Document Representations Using Natural Language Processing

I have no dictionary, and I do not want one;
I can select words by the sound, or by orthographic aspect.
Many of them have French or German or English look, and these are
the ones I enslave for the day's service. That is, as a rule.
Not always. If I find a learnable phrase that has an imposing look
and warbles musically along I do not care to know the meaning of it;
I pay it out to the first applicant, knowing that
if I pronounce it carefully he will understand it,
and that's enough.

Mark Twain (*Italian without a Master*, 1904)

Text processing programs can often be as ignorant of language in general as Mark Twain pretended to be of the idiom spoken in Italy. Nobody cares as long as the desired effect is achieved. But just as the famous writer might eventually have found communication even smoother with proper knowledge of the Italian language, the question must be asked whether and to what extent users may profit if their text processing programs are enriched by an examination of the words and symbols beyond "the sound, or the orthographic aspect".

The present chapter describes a number of relatively simple document representation methods. They all have in common that the representation is *reduced* in one way or another, either by omitting certain features or by mapping them onto some standardised form. In our earlier terminology (Section 3.4) these can be feature *selection*, *standardisation* or even *extraction* methods. The next chapter is then devoted to more demanding approaches which mostly aim to *enhance* the feature space by way of *feature extraction*. In both chapters we have a particular eye on linguistically motivated techniques. All are evaluated on the basis of our five German data sets.

5.1 Baseline

In order to evaluate the impact of feature representation techniques, a *baseline* needs to be determined—the set of results that can be achieved by simple standard means. All further experiments can then be measured against this baseline.

5.1.1 Preparation

Following common practice, the simple bag-of-words model was chosen as a baseline. **Vectorisation** was performed with the tokenising module coming as part of the GERTWOL software (cf. Section 5.2.1 below).

A number of refinements typically encountered in practice was then calculated on this simple BOW model:

1. Removal of punctuation, numbers and other "non-alphabetic" tokens (here defined as tokens with less than two ordinary letters).

2. Removal of stopwords (see Section 5.4 for more details).

3. Stemming of all tokens with Porter's German stemmer, including conversion to lower case.[1]

The results (Figure 5.1/Table D.5) are in themselves quite interesting in that they only partially support the prevailing view that stopword removal and stemming automatically lead to better results. As a matter of fact, the impact of these methods on the NZZ corpus appears to be negligible while the AMAZON and WIKI corpora show even better results if stopping is *omitted*. We will return to these anomalies further below.[2]

5.1.2 Interpretation

From these experiments we can construct two baselines:

	SPRINGER	AMAZON	SDA	WIKI	NZZ
Baseline 1 ("standard") [= $BOW_{stop,\ stem}$]	0.410[0.010]	0.468[0.006]	0.431[0.002]	0.416[0.019]	0.348[0.002]
Baseline 2 ("best")	0.408[0.004]	0.458[0.009]	0.431[0.002]	0.386[0.010]	0.346[0.001]

Baseline 1 shows the results achieved by the state-of-the-art, non-linguistic feature representation method (stopping and stemming). It is the baseline usually referred to in the further text, even though from the results in Figure 5.1 it appears debatable whether or not stopword removal should actually be included. *Baseline 2* is constructed by individually taking the best

[1] Porter's German stemmer is available from `www.snowball.tartarus.org/algorithms/german/stemmer.html`.

[2] Curiously, straightforward improvements for these techniques and their combination can be observed in all five data sets if the double-weighting method (Section 4.4.2.1) is given up in favour of the (less powerful, but standard) LOG-IDF method:

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOW	0.627[0.045]	0.552[0.007]	0.551[0.001]	0.626[0.019]	0.452[0.018]
BOW_{stop}	0.536[0.017]	0.507[0.004]	0.523[0.000]	0.427[0.008]	0.412[0.020]
BOW_{stem}	0.514[0.013]	0.533[0.004]	0.532[0.000]	0.531[0.020]	0.433[0.021]
$BOW_{stop,\ stem}$	0.469[0.010]	0.500[0.005]	0.522[0.000]	0.404[0.008]	0.398[0.001]

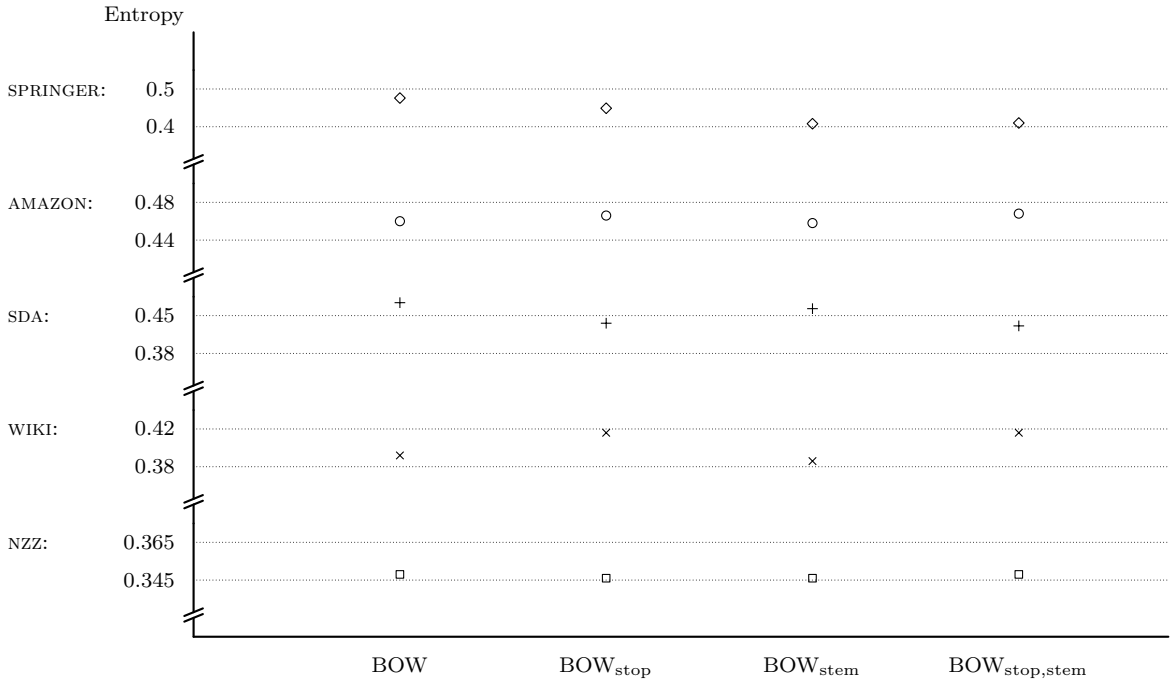


Figure 5.1: **Baseline and variants.** BOW = standard bag-of-words; BOW$_{stop}$ = bag-of-words after stopword removal; BOW$_{stop,\ stem}$ = bag-of-words after stopword removal and stemming. (For the underlying numbers refer to Table D.5.)

result from Table D.5 for each data set. This baseline is more difficult and more desirable to beat, but less representative since the methods were picked *ex post* for each set.

To illustrate the cluster distributions under *Baseline 1*, Tables 5.1 to 5.5 give the respective confusion matrices of a typical run in each data set. It can be observed that for the sets with many categories (AMAZON and WIKI) there is no perfect 1:1 match between categories and clusters; categories with very many documents are spread over several clusters and so are categories with very few members. Interestingly, a similar behaviour can be detected in the NZZ set, where the FEUI (Feuilleton) category is distributed over two big clusters whereas INLA (Inland) and ZURI (Zürich) are merged. The cluster algorithm thus detects a different principal structure—and it actually happens also to correspond to a reality: in the physical archive of the *Neue Zürcher Zeitung* no distinction is made between the two departments Inland and Zürich as the two are closely interconnected.

Even if the 1:1 relation between clusters and labels appears thus to be broken in various instances, the entropy measure still manages to provide a sensible and sensitive assessment of the degree of correspondence between clusters and labels.

On the whole, the baseline clustering results may appear to be relatively good. Expressed in terms of the "classification error" (Eq. 2.71), however, we still have error rates of 24.7% (SPRINGER), 44.7% (AMAZON), 27.3% (SDA), 39.0% (WIKI) and 21.0% (NZZ).

	TECH	BWLM	MEDI	INFO	ARCH	JURA	VOLK
C_1	378	50	11	23	0	9	47
C_2	101	367	14	51	5	3	13
C_3	3	1	993	2	34	6	0
C_4	85	43	17	246	1	0	5
C_5	38	11	47	12	214	9	4
C_6	13	13	20	3	0	461	1
C_7	38	130	46	7	4	28	229

Table 5.1: **Confusion matrix of the** SPRINGER **baseline 1 result** (classification error: 24.7%).

	INLA	AUSL	KULT	VERM	WIRT
C_1	15516	89	150	674	1401
C_2	201	11104	19	1009	51
C_3	2355	5061	4311	4779	1912
C_4	478	163	21	10271	25
C_5	259	54	10	61	8708

Table 5.2: **Confusion matrix of the** SDA **baseline 1 result** (classification error: 27.3%).

	INLA	VERM	SPOR	FEUI	AUSL	WIRT	ZURI
C_1	3031	173	37	32	104	310	1904
C_2	220	4156	43	346	1259	36	384
C_3	0	36	3125	4	0	0	25
C_4	7	401	2	2442	3	1	270
C_5	229	533	21	5514	229	33	277
C_6	140	67	1	74	7023	86	1
C_7	118	14	0	8	64	3040	38

Table 5.3: **Confusion matrix of the** NZZ **baseline 1 result** (classification error: 21.0%).

	KIND	ARCH	RATG	KULT	BELL	BIOG	RELI	BUSI	HIST	BIOC	GERM	INGE	COMP	MATH	SPOR	SCIF	EROS	KUNS	REIS	LIFE	PUBL
C_1	4061	0	32	8	114	38	66	3	10	37	9	10	10	0	30	1	1	4	5	31	0
C_2	2512	0	5	6	49	20	2	1	3	1	4	0	0	0	14	1	0	0	2	0	0
C_3	1596	0	18	7	162	32	177	0	4	2	20	1	0	0	8	3	4	0	4	4	1
C_4	1501	14	14	30	915	262	22	3	75	97	21	4	10	0	171	141	3	7	106	18	3
C_5	79	1	1737	7	50	120	21	15	16	139	11	55	7	1	168	0	0	3	4	103	1
C_6	119	177	28	1041	122	547	37	24	87	41	169	47	14	0	74	49	1	420	52	95	174
C_7	2737	1	114	83	4387	473	26	21	24	6	22	7	4	0	16	193	120	1	2	7	1
C_8	256	2	83	48	3900	936	44	22	74	40	82	15	5	1	44	38	181	9	3	38	41
C_9	364	2	5	21	2627	62	8	3	3	0	7	2	1	0	1	11	6	0	0	0	0
C_{10}	269	8	31	63	579	312	72	9	72	24	31	1	6	1	34	1	7	15	10	7	32
C_{11}	230	18	27	57	1000	1743	427	17	692	6	40	12	3	0	13	0	1	20	4	8	16
C_{12}	143	13	12	41	887	1100	22	4	80	16	291	3	2	2	10	2	36	15	2	15	24
C_{13}	39	4	14	3	52	72	1014	2	46	0	7	1	0	0	12	0	0	4	16	17	1
C_{14}	166	5	23	2	110	373	1551	4	60	7	37	2	1	0	8	3	0	3	2	6	4
C_{15}	157	13	1206	117	57	58	34	1490	72	268	82	143	467	11	910	3	8	16	19	308	56
C_{16}	52	10	69	13	10	33	14	106	122	1244	359	211	131	629	73	0	0	4	1	36	33
C_{17}	16	298	13	37	0	7	3	168	11	82	8	1772	62	5	52	0	0	14	42	23	24
C_{18}	13	0	12	31	6	6	1	40	12	24	21	58	1201	15	16	1	0	4	2	8	9
C_{19}	51	39	395	138	60	52	27	68	52	1851	46	725	190	2067	64	1	0	80	5	10	10
C_{20}	47	10	30	2	15	44	4	2	12	102	31	35	4	10	958	0	0	1	40	36	3
C_{21}	328	11	28	29	310	129	21	16	52	17	25	22	20	2	514	6	1	8	2	13	5

Table 5.4: **Confusion matrix of the AMAZON baseline 1 result** (classification error: 44.7%)

	HIST	INFO	MATH	SPRA	PHIL	KUNS	LITE	BUSI	BIOL	FREI	MEDZ	TECH	SPOR	RELI	JURA	LIFE	ASTR	PHYS	ORGA	CHEM	SEXU	SOZI
C_1	1664	1	0	19	4	8	44	26	13	80	1	3	20	28	21	2	1	0	56	0	3	7
C_2	1375	0	1	21	1	4	34	1	3	4	2	0	1	326	3	0	1	0	0	0	1	11
C_3	945	0	0	399	0	1	2	0	0	0	0	0	0	267	2	0	0	0	9	0	0	3
C_4	857	0	4	74	6	57	54	16	12	3	28	9	12	26	0	0	5	59	4	16	0	7
C_5	719	7	60	130	53	104	281	432	110	32	56	59	36	138	98	17	79	142	313	58	10	138
C_6	5	1781	14	495	1	52	61	128	1	281	2	436	0	5	34	2	1	4	12	3	0	33
C_7	19	50	787	999	1002	20	51	15	21	39	4	12	32	62	4	0	10	20	3	0	4	84
C_8	9	0	0	15	0	1918	18	10	1	17	0	4	5	2	2	1	0	0	1	0	1	6
C_9	11	1	1	20	2	1089	254	21	4	91	5	486	14	8	2	3	1	0	5	0	10	18
C_{10}	10	0	2	15	1	1075	49	5	7	10	5	8	17	6	0	1	0	5	1	0	2	5
C_{11}	52	3	10	59	32	305	3059	23	22	93	7	8	7	69	3	1	7	2	6	6	4	52
C_{12}	222	2	1	84	0	1	3	2003	23	48	0	121	51	2	5	1	12	6	21	1	0	4
C_{13}	29	19	1	46	2	4	4	1286	488	38	5	16	20	3	3	1052	1	1	7	22	0	7
C_{14}	19	1	0	5	0	2	6	50	4463	67	18	3	10	3	1	4	9	0	1	8	1	6
C_{15}	212	5	24	51	59	8	33	1003	281	61	1227	51	56	158	889	9	1	9	214	33	252	220
C_{16}	21	64	9	51	3	73	2	205	38	30	30	2176	63	7	2	5	36	221	56	26	2	53
C_{17}	8	1	4	28	1	3	2	237	19	133	0	2	2286	0	0	0	0	0	2	0	3	1
C_{18}	44	0	1	26	9	14	37	1	3	72	0	3	0	941	7	1	0	0	49	0	5	10
C_{19}	52	0	0	21	0	49	42	18	5	60	2	1	1	938	1	5	1	1	1	0	0	4
C_{20}	184	1	16	295	51	70	269	15	28	33	8	7	14	748	13	3	9	0	6	6	23	34
C_{21}	0	0	0	220	0	1	2	1	279	0	0	0	0	1	0	0	1436	2	0	0	0	0
C_{22}	26	6	444	13	12	7	4	56	339	5	80	142	6	2	1	25	259	652	19	1207	0	3

Table 5.5: **Confusion matrix of the WIKI baseline 1 result** (classification error: 39.0%).

5.2 Bag-of-Lemmata

The *bag-of-lemmata* (BOL) is the starting point of several higher-level feature refinement methods. In order to obtain a BOL representation of our documents (instead of a BOW model), the documents must be syntactically analysed before vectorisation (cf. Section 3.2.2).

5.2.1 POS Tagging and Lemmatising

POS tagging and lemmatising was performed in two separate steps:

1. For POS tagging we used Helmut Schmid's TREE-TAGGER (cf. Schmid, 1994, 1999).
2. For lemmatising and morphological analysis we relied on the German morphological analyser GERTWOL (cf. Haapalainen and Majorin, 1995).

If POS tagging failed or resulted in an unknown tag, we used the POS information generated by GERTWOL. If more than one lemma was available, Volk's adapted algorithm for determining the most probable lemma for adjectives, verbs and nouns was used (Volk, 1999). Similarly, if GERTWOL failed to analyse a word, the lemma returned by the TREE-TAGGER was used.[3]

For our purposes the respective tag sets of the two programs were mapped onto a simplified generic scheme with 15 POS categories as in Table 5.6.

Table 5.6: **Generic mapping of the Tree-Tagger and Gertwol tag sets.**

Generic tag	TREE-TAGGER	GERTWOL	
ADJ	ADJA ADJD	A	*Adjectives.*
ADV	ADV	ADV	*Adverbs.*
APP	APPO APPR APPRART APZR	PRÄP	*Appositions (prepositions, postpositions, etc.).*
ART	ART	ART DET	*Determiners.*
KON	KOKOM KON KOUI KOUS	KONJ	*Conjunctions (coordinating, subordinating, etc.).*
NAM	NE	EIGEN	*Proper names.*
NAM_1	[NE]	EIGEN Vorname	*Proper names, recognised by* GERTWOL *as first names.*
NAM_2	[NE]	EIGEN Famname	*Proper names, recognised by* GERTWOL *as family names.*
NAM_{all}			*Used for the union of* NAM, NAM_1 *and* NAM_2.

[3] In order to avoid confusion further on, it should be noted that GERTWOL always resolves contracted prepositions such as "beim" (→ "bei-der"), "zur" (→ "zu-der") or "im" (→ "in-der").

Table 5.6: Generic mapping of TREE-TAGGER and GERTWOL tag sets.

Generic tag	TREE-TAGGER	GERTWOL	
NUM	CARD	NUM KARD ORD BRUCH RÖM	*Numbers of all sorts.*
PRO	PAV PDAT PDS PIAT PIDAT PIS PPER PPOSAT PPOSS PRELAT PRELS PRF PWAT PWAV PWS	PRON PRONADV	*Pronouns of all colours (since they all have little interest for clustering, no further distinction is made).*
PTK	ITJ PTKA PTKANT PTKNEG PTKVZ PTKZU PTKZV	INTERJ PRÄF	*Particles and isolated prefixes.*
PUN	$($, $.	KOMMA PUNKT DOPPELPUNKT VKLAMMER HKLAMMER FRAGMENT PROZENTZEICHEN FRAGEZEICHEN SEMIKOLON SCHRÄGSTRICH AUSRUFEZEICHEN BINDESTRICH ANFÜHRUNGSZEICHEN GEDANKENSTRICH PARAGRAPHZEICHEN STERN-ZEICHEN PLUS-ZEICHEN MINUS-ZEICHEN	*Punctuation marks.*

Table 5.6: Generic mapping of TREE-TAGGER and GERTWOL tag sets.

Generic tag	TREE-TAGGER	GERTWOL	
		GLEICHHEITSZEICHEN NUMMERNZEICHEN ET-ZEICHEN AUSLASSUNGSPUNKTE	
SUB	NN	S	*Nouns.*
UNK	FM TRUNC XY	ERSTGLIED	*Unknown tokens, often first elements from complex expressions (e. g. the term "Berg-" in "Berg- und Talfahrt").*
VRB	VAFIN VAIMP VAINF VAPP VMFIN VMINF VMPP VVFIN VVIMP VVINF VVIZU VVPP	V	*Verbs.*

5.2.2 Experimental Results

Is the BOL representation more suitable than BOW? Figure 5.2 seems to indicate so. Disregarding the NZZ corpus, which shows only minimal variation, we find BOL to be better than BOW on each occasion. However, if stemming is added to the equation it transpires that on the whole there are no big differences between stemming and the less radical lemmatising approach. Combining lemmatising and stemming (BOL_{stem}) does not seem to have much (positive) impact.

Conclusions. Lemmatising improves results if compared to simple BOW. On its own, the extra effort invested in lemmatising does not pay off, however, as the alternative, stemming, offers a similar effect at a much lower cost. The tendency towards over-generalisation inherent to stemming does not seem to be a disadvantage. In terms of dimensions (features), lemmatising reduced the feature space in our experiments by 15–23 percent, whereas stemming led to a reduction of 25–30 percent. At the cost of increasing pre-processing time, both techniques help to reduce computational demands in the clustering phase.

Even though tagging/lemmatising does not show better results than stemming, it may nevertheless be useful if the extra information serves as input for the more sophisticated methods discussed further below in this and the next chapter.

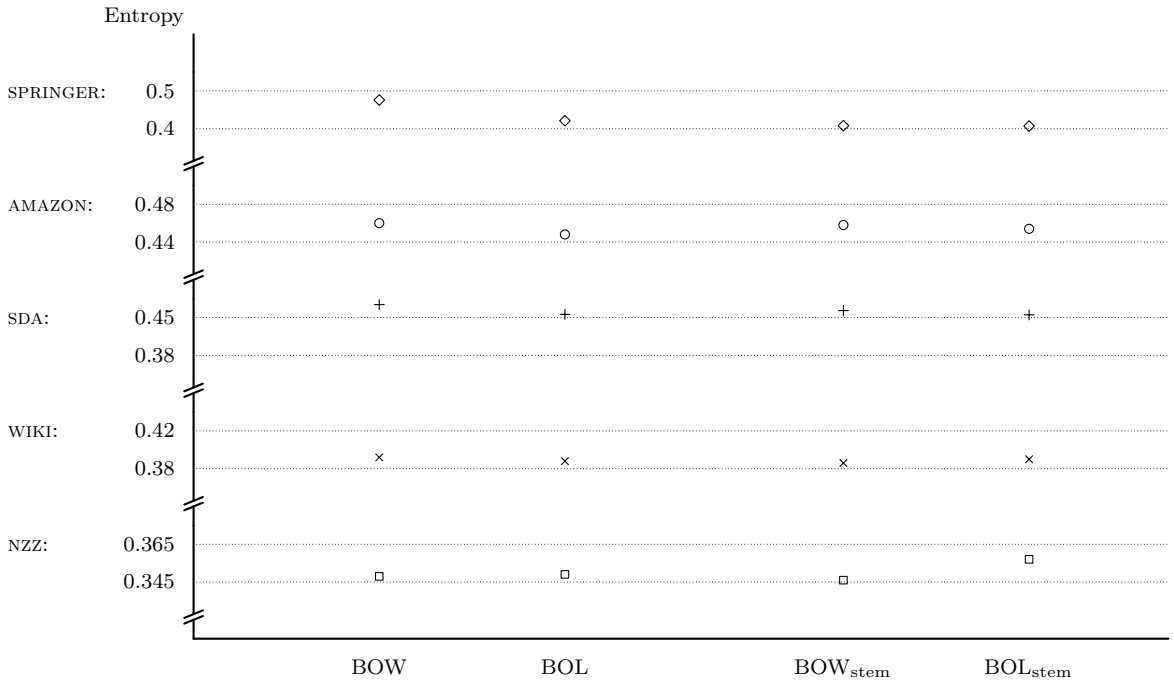


Figure 5.2: **Bag-of-lemmata versus bag-of-words.** (For the underlying numbers refer to Table D.6.)

5.3 Statistical Reduction Techniques

This and the next four sections deal with matrix size reduction.

In the present section matrix reduction techniques are examined that do completely without any knowledge of the language and rely solely on the given data. Models that make use of additional (linguistic) knowledge are dealt with in the sections that follow.

Reducing the feature matrix for clustering can be done with two separate and sometimes opposing goals:

- *Qualitative improvement:* Matrix reduction as a means of improving quality by reducing "noise", i. e. by removing data that is more likely to obscure the underlying cluster structure than to reveal it.
- *Quantitative improvement:* Matrix reduction as a means of reducing space and time requirements of the clustering process, or allowing a larger number of documents to be clustered under equal space and time restrictions.

We measure the two aspects using the methodology outlined in Section 4.3.3.

5.3.1 Pruning

We examine three approaches to matrix pruning:

Global pruning with similarity preservation. The *–colprune=*$\alpha/100$ parameter prompts CLUTO to remove all features which are not necessary to account for α percent of the overall similarity between documents. According to the manual, substantial dimensionality reductions are thus possible without seriously affecting clustering quality. It recommends to set α between 80 and 100 (the latter means no pruning at all and is the default value).

Global pruning with upper and lower bounds. As an alternative we examined the exclusion of very high- and low-frequency words using upper and lower bounds on the *document frequency* of the features. Upper bounds between 0.5 and 10 percent and lower bounds between 0 and 0.1 percent were tested.

Local pruning. Finally we evaluated a local pruning technique, reducing each document to its α most frequent terms. The procedure was tested in two variations: at first with all features available, and then with only those features that actually occur in more than one document. Since the fraction of these "shared" features is only between 34 and 45 percent of all features (but accounting for 90 to 98 percent of the non-zero elements!) the results are expected to be different. The advantage of the first method is that it can be implemented already at vectorisation stage, whereas the second is only possible at matrix assembly time.

All experiments used the BOL model. The results are shown in Figures 5.3, 5.4 and 5.5, whereas Table 5.7 indicates the time used by CLUTO to process the different input matrices (at the example of the SDA data set[4]).

We find that if applied very moderately ($\alpha = 99$) *pruning with similarity preservation* (Figure 5.3) gives qualitatively good results for three of the five data sets, while considerably saving time (ca. 25% for the SDA data set). Larger pruning factors cause the results to deteriorate, even though the additional time gains are rather substantial.

[4] As we are only interested in the relative time requirements, evaluation on a single data set is enough for our purposes. For the specifics of the machine used for these experiments refer to the description on page 99.

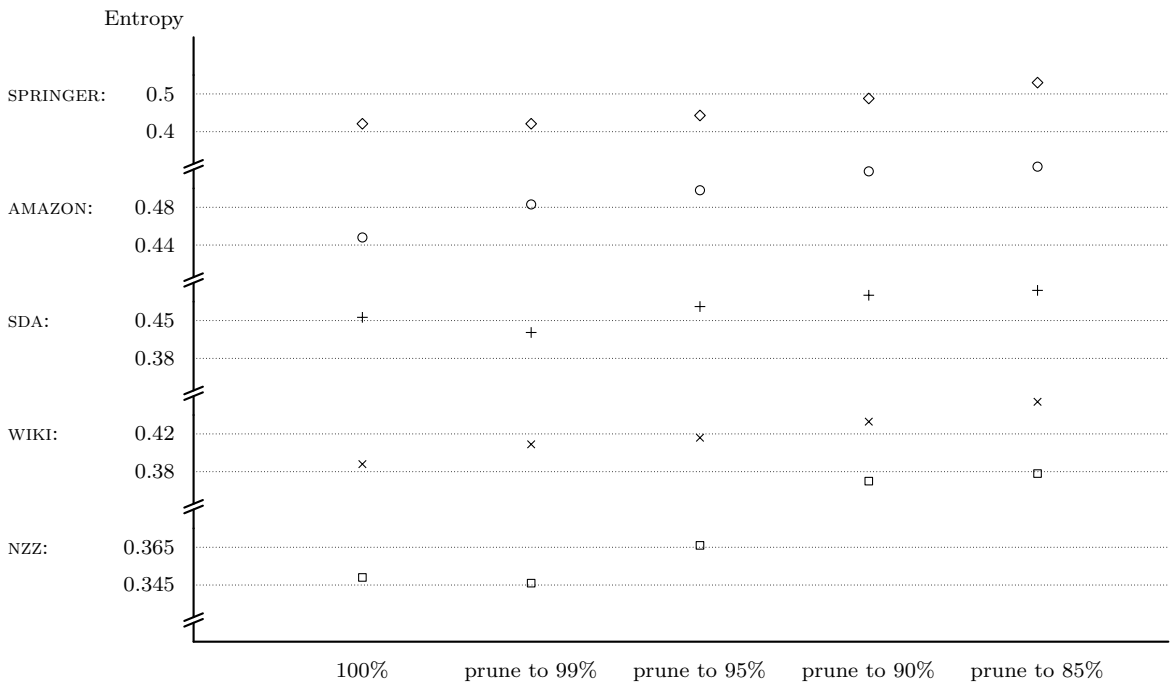


Figure 5.3: **Global pruning with similarity preservation,** using CLUTO's *–colprune* parameter. (For the underlying numbers refer to Table D.7.)

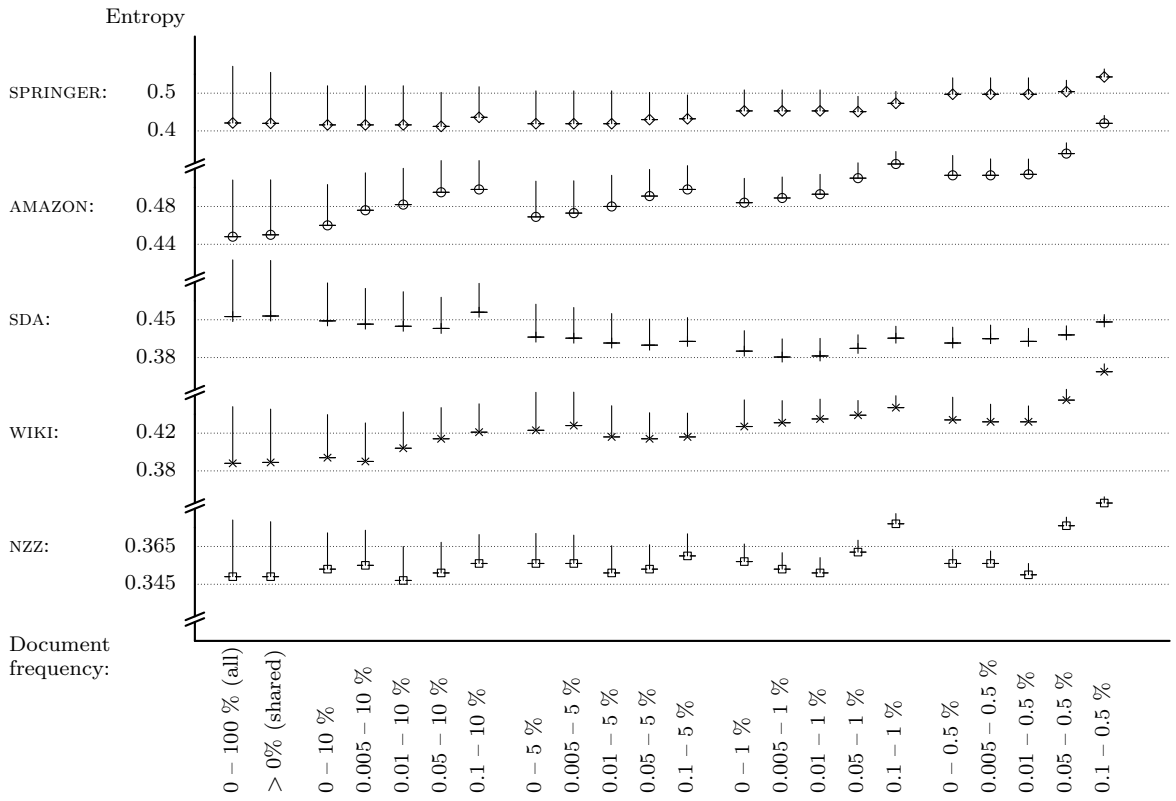


Figure 5.4: **Global pruning with upper and lower bounds** on document frequency. The length of the vertical lines on top of each data point indicate the number of non-zero elements in the resulting feature-document matrix. The shorter the line, the smaller the matrix and the less time is needed for clustering. (For the underlying numbers refer to Table D.8.)

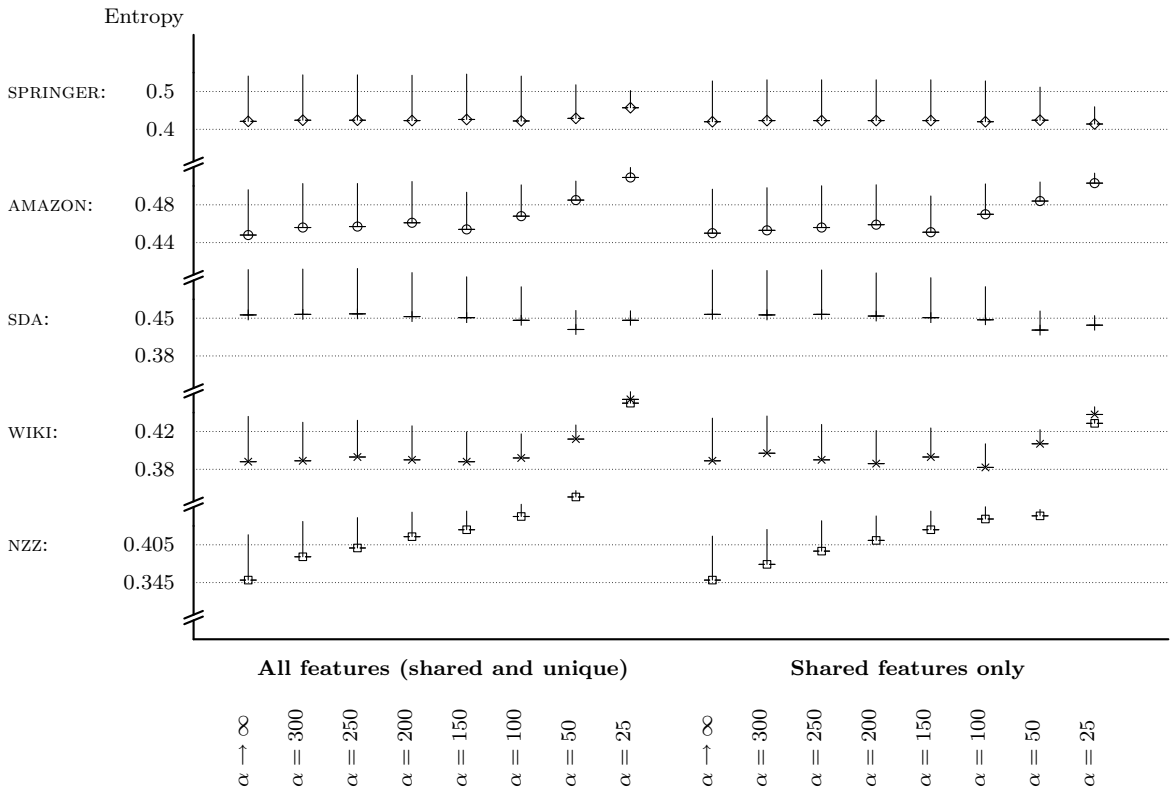


Figure 5.5: **Local pruning:** keeping only the α most frequent features of each document (after LOG-IDF weighting). The left half of the figure shows the procedure with all features. In the right half those features not occurring in other documents ("unique features") were removed before making the selection. The differences are not big except for the cases where documents would have been lost altogether.
Note: *The scale is not the same for each data set, nor is it always the same between different figures; here, for instance, the* NZZ *scale is different than in the previous figure. Generally, though, it was attempted not to switch too often.*
The length of the vertical lines on top of each data point indicate the number of non-zero elements in the resulting feature-document matrix. The shorter the line, the smaller the matrix and the less time is needed for clustering. (For the underlying numbers refer to Table D.9.)

The results gained with *lower and upper bounds on document frequency* (Figure 5.4) are less straightforward to interpret. Most striking is the fact that while the other data sets show a gradual decline in the cluster results with shrinking matrix size, the SDA results behave in the opposite way: the more we reduce the available features, the better the clustering! The best result actually occurs with the 0.005–1% range which retains as few as 32% of the non-zero elements of the SDA input matrix. But also for the other data sets we find that substantial quantitative reductions are possible at quite small costs in clustering quality. In particular it stands out that 20 percent of the NZZ set (the 0.1–0.5% range) can lead to comparable results as the full set, whereas for SPRINGER and WIKI an upper barrier of 10% document frequency helps reducing the matrix size by 30% or more with little or no loss of quality. We also note, however, that for the AMAZON set all pruning leads to an immediate loss in quality.[5]

From these experiments it is not possible to deduce more specific recommendations than to generally observe that both a generous upper bound (10%) and a relatively small lower bound (excluding terms occurring in less than 5–10 documents) appear to be useful under many circumstances.

Local feature pruning strategies (Figure 5.5) show roughly the same picture as global strategies, but the effects are less severe even when the matrix is much reduced. Again a qualitative improvement for the SDA set can be observed (except for $\alpha = 10$), while the SPRINGER set profits at least quantitatively. The WIKI set responds unexpectedly well even to considerable reductions, while the AMAZON set shows good results for the higher α values. Finally, the NZZ results suffer most from local reduction, which is not too surprising given that the documents are much longer on average than the others, which means that with increasing document length the stopwords tend to occur and be repeated more often than the typical content-words (which in a longer text are rarely constantly present from beginning to end).

5.3.2 Latent Semantic Analysis

Separately, we briefly tested Latent Semantic Analysis (see Section 3.4.3.2) as a matrix reduction technique. Using *Singular Value Decomposition* the sparse, high dimensional document matrix was projected into compact subspaces of $\rho = 5 \ldots 300$ feature dimensions, i. e. the original $n \times m$ matrix H was approximated by $H \approx US_\rho V^T$ and the new $n \times \rho$ matrix $H' = US_\rho$ was used for clustering.

It turned out that the (quadratic) memory demands of Singular Value Decomposition exceeded our computational capabilities for all except the SPRINGER data set by far[6] (cf. Table D.10). Since those results that were available could not be described as very encouraging either, LSA was regarded as too expensive and success too uncertain to warrant further investigations in this context.[7]

5.3.3 Conclusions

Whereas our LSA experiments proved inconclusive, it could be shown that feature pruning is potentially a very useful technique, both from a quantitative and qualitative point of view.

[5] In order to put the entropy decreases into perspective, compare with the results that we found prior to applying our "external" extra weighting scheme (Figure 4.8). It then transpires that even though the AMAZON results get immediately worse with pruning, it is still possible to more than halve the number of non-zero elements and still achieve a better clustering result than with any of CLUTO's native global weighting strategies.

[6] The bag-of-lemmata being used after external weighting and retaining only shared features.

[7] SVD was performed with 1 GB RAM and with MATLAB's SVDS routine which is based on the LAPACK package (Anderson *et al.*, 1999). Attempts with the SVDPACKC package (Berry *et al.*, 1993) did not solve the computational difficulties.

method	*retained*	*time*	*lost*
No pruning (100 %; $\alpha \to \infty$)			
complete feature matrix	100%	218s	
Global pruning with similarity preservation[a]			
prune to 99%		162s	
prune to 95%		127s	
prune to 90%		111s	
prune to 85%		104s	
Global pruning, upper and lower bounds			
0 – 10 %	67%	153s	
0.005 – 10 %	63%	134s	
0.01 – 10 %	61%	125s	
0.05 – 10 %	55%	99s	
0.1 – 10 %	51%	89s	
0 – 5 %	58%	144s	
0.005 – 5 %	54%	123s	
0.01 – 5 %	52%	116s	
0.05 – 5 %	46%	90s	
0.1 – 5 %	42%	75s	
0 – 1 %	36%	112s	
0.005 – 1 %	32%	91s	
0.01 – 1 %	31%	84s	
0.05 – 1 %	24%	59s	{2}
0.1 – 1 %	21%	50s	{7}
0 – 0.5 %	28%	101s	
0.005 – 0.5 %	24%	77s	{1}
0.01 – 0.5 %	23%	70s	{1}
0.05 – 0.5 %	16%	49s	{15}
0.1 – 0.5 %	13%	40s	{56}
Local pruning (all features)			
$\alpha = 300$	100%	189s	
$\alpha = 250$	100%	189s	
$\alpha = 200$	97%	182s	
$\alpha = 150$	90%	177s	
$\alpha = 100$	74%	157s	
$\alpha = 50$	42%	120s	
$\alpha = 25$	21%	79s	
$\alpha = 10$	8%	43s	{83}

[a]Times include pruning process.

Table 5.7: **Time demands of different pruning techniques** for the SDA data set (*retained:* percentage of non-zeroes from input matrix kept after pruning; *time:* duration needed by CLUTO for clustering, including I/O-processing and reporting time; *lost:* number of documents that could not be clustered because no feature survived pruning).

However, it was also shown that the benefits to be gained from pruning are strongly dependent on the concrete task (data).

From a qualitative perspective, it was surprising to find that one data set (SDA) derived great benefits from practically all pruning methods, even very severe ones, whereas for other sets (NZZ and in particular AMAZON) it turned out to be nearly impossible to improve clustering quality by way of matrix pruning.[8]

From a quantitative perspective, it is encouraging that it appears possible to reduce matrix size from 20% (AMAZON) up to 80% (SDA and NZZ) without qualitative loss. Based on the present data, it is not possible though to indicate definitely which pruning method and which parameters to use in which situation.

Global pruning with feature preservation quickly led to a decline in the results. Global pruning with upper and lower bounds performed well for SDA and NZZ, while for AMAZON and WIKI local pruning seemed preferable.

Generally speaking, pruning is easier if the underlying categorical structure (class labels) is crude. The SPRINGER, SDA and NZZ sets come with only 5 or 7 classes and have presumably more redundant information that can be stripped before clustering. The AMAZON and WIKI sets with over 20 classes are more dependent on the individual feature and pruning is more likely to remove important information. As a tentative conclusion we suggest that for larger and more detailed clusterings local pruning is preferable, while for coarse clustering task a generous global pruning strategy with lower and upper bounds can be used.

In the following sections we examine whether language knowledge can help us identify beneficial reductions with greater certainty than has been the case with a strictly statistical approach.

5.4 Stopwords

Stopword lists are one of the oldest complexity reduction techniques and are resorted to in practically all current IR systems. Their effectiveness seems to be unquestionable. Under these circumstances it is a bit surprising that comparatively little research has gone into exploring the nature of stopwords. Length and content of stoplists can vary strongly and there is no standard procedure for creating them. Which word to consider a stopword indeed depends often on the actual application and context.

Since we know of no universally accepted stopword list in *German*, we manually compiled a list of our own, drawing from various sources. The list has 1417 entries stemming from ca. 800 lemmata. In addition we used a specific stopword list of 588 *English* words and word forms (including a few Roman numerals and typical HTML abbreviations) because various texts were found to contain English text strings. Both lists are given in Appendix C.

In the present section we will first look at the influence of our stoplist on cluster results. In a second step we evaluate a number of automated stopword extraction techniques. Note that all experiments outside this section use the initial, manually compiled lists of Appendix C.

[8] Since the results might lead one to believe that *any* pruning would have been good for SDA, no matter how the reduction was chosen, we performed an additional experiment wherein we removed every second column from the matrix. In accordance with our expectations, the results were *worse* than with the systematic pruning methods. It was a surprise, however, to find the NZZ results being quite stable even with this random pruning method (remember that numbers in parenthesis refer not to dimensions but to non-zero elements and may therefore diverge from 50%):

	SPRINGER	AMAZON	SDA	WIKI	NZZ
no reduction	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
50% random reduction	0.500 (49%)	0.510 (49%)	0.474 (48%)	0.460 (49%)	0.352 (51%)

5.4.1 Explicit Stoplists

Figure 5.6 reports on the stopword experiments, with our manual stoplist applied both to the BOW and BOL models. The results are quite in accordance with our findings in the previous section: small qualitative impact on SPRINGER and NZZ, negative consequences for AMAZON and WIKI, and the only clear improvement for the SDA set. This evidence forces us to conclude that—contrary to wide-spread practice—*stopword removal cannot be universally recommended for all document clustering tasks.*

As it may be argued that perhaps our rather long and comprehensive stoplist is responsible for these counter-intuitive results, the experiment was repeated with a much smaller stoplist, i. e. the stopwords used by Google.[9] This alternative list is restricted to 127 German and 35 English words. It turns out that the situation does not change much. The direction of the effects remained the same, even though on a smaller scale (not unexpectedly).

5.4.2 Stopword Extraction Techniques

In view of the surprisingly mixed experiences just reported, the question arises whether human intuitions about stopwords are perhaps inaccurate and whether the task of stoplist generation could, and perhaps should, be automated. In order to explore this subject we tested several measures in a classification context, with the aim of gauging the "stopwordliness" of individual words/lemmata.

Intuitively speaking, a good stopword should appear frequently enough to be worthwhile removing and be unrelated to the document content. This content-independence is here crudely reflected in the distribution of class labels. Taking the BOL model as our base and partly inspired by research in the domains of corpora comparison (Kilgariff, 1997, 2001; Roeck *et al.*, 2004a,b) as well as text categorisation (Wilbur and Sirotkin, 1992; Yang and Pedersen, 1997; Pekar *et al.*, 2004), we tested eight techniques to assess a term's "stopwordliness" within a given labelled corpus.

We start with a few preliminary definitions for a given feature f_i and label L_j:

- A = the number of documents containing f_i and having label L_j,
- B = the number of documents containing f_i but having a label different from L_j,
- C = the number of documents not containing f_i and having label L_j,
- D = the number of documents neither containing F_j nor having label L_i,
- $p_1 = A/(A+C)$ = the fraction of documents with label L_j that contain f_i,
- $p_2 = A/(A+B)$ = the fraction of documents containing f_i that have label L_j,
- $p_3 = \frac{A/(A+C)}{(A/(A+C))+(B/(B+D))}$,
- $p_4 = \frac{A/(A+B)}{(A/(A+B))+(B/(B+D))}$.

We can then define the following measures of stopwordliness $s(f_i)$:

Document Frequency. One of the simplest stopword detection criteria is document frequency. It assumes that the more often a word occurs, the less content it bears. Stopwordliness is thus measured simply by $s(f_i) = \sum A$.

[9] The Google stopwords were taken from the empirical list given at `www.ranks.nl/stopwords/`.

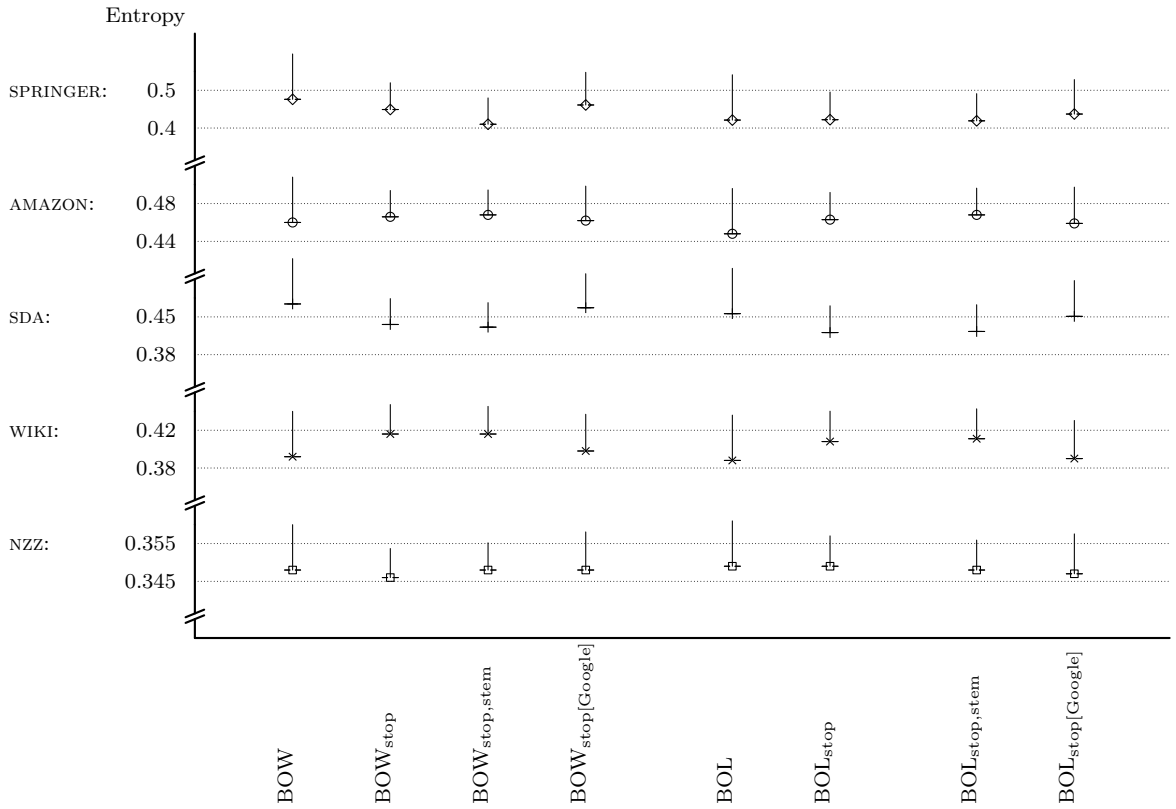


Figure 5.6: **Stopword removal** with a manually compiled stoplist of around 800 German lemmata and 588 English word forms. As a reference the results with the much shorter "Google stoplist" (see text) are also given in the fourth and eighth column. (For the underlying numbers refer to Table D.11.)

df	**discr**	χ^2	**E**	**E'**	**E"**	**KL**	**WKL**
der	der	der	Menschenverstand	spendabel	erst	der	der
die	die	anlangen	Vizevorsitzend	diagnostiziert	der	die	die
ein	ein	Auftraggeber	zweischneidig	Geldgeschäft	Freitag	durch	ein
in	und	unwahr	Autobiographie	wund	drei	ein	und
und	durch	wasser-	Herstellen	Weichmacher	das	und	in
sein	sein	miserabel	zerstreut	Urteilsverkündigung	Dienstag	sein	sein
werden	in	Chemiker	einbrocken	Greif	Donnerstag	in	er
an-der	dann	Filter	Fachkonferenz	Sicherheitsprüfung	Mittwoch	dann	werden
haben	werden	dumm	zusammenschweißen	aufrütteln	durch	er	durch
von	von	öfters	Geschäftsidee	Stippvisite	weiter	werden	an-der

Table 5.8: **Top ten stopwords** as found by eight measures in the labelled SDA corpus.

Discriminative Power. Following Liu *et al.* (2002) we use discriminative power to establish how much each term contributes to a specific category. We use the following definition of discriminative power, based on relative document frequencies: $discr(f_i) = \log \frac{\max p_1}{(\sum p_1 - \max p_1)/(k-1)}$. Our measure then becomes $s(f_i) = 1 - discr(f_i)$.

Chi-Square Test (χ^2). The chi-square test is a typical measure to evaluate how much a certain term contributes towards explaining a certain classification. Here we want to use it to identify terms that do *not* contribute significantly to any of the categories. The χ^2 statistic is defined[10] as $\sum_{L_1 \ldots L_k} (O-E)^2/E$ where the *observed* value $O = A$ and the *expected* value $E = (A+B)(A+C)/n$. We then calculate $s(f_i) = 1 - \chi^2(f_i)$.

Entropy. The standard entropy of a term's distribution is $s(f_i) = E = \frac{-1}{\log n} \sum p_2 \log p_2$.

E'. Following Xiao (2003) we define E' as $E'(f_i, L_j) = p_1(1 + p_3 \log_2 p_3 + q \log_2 q)$, with $q = 1 - p_3$. The smaller E', the less informative is the feature about L_j, and thus $s(f_i) = 1 - \max_{L_1 \ldots L_k} E'(f_i, L_j)$.

E''. The empirical measure E'' is identical to E' except for the replacement of p_3 by p_4.

Kullback-Leibler Divergence (Relative Entropy). The Kullback-Leibler divergence measures the "distance" between two distributions (the distribution of all documents versus the distribution of documents containing f_i). With $q = (A+C)/n$ it is defined as $KL(f_i) = \sum p_1 \log \frac{p_1}{q}$. Stopwordliness can then be measured as $s(f_i) = 1 - KL(f_i)$.

Weighted Kullback-Leibler Divergence. Finally, a weighted version of the Kullback-Leibler divergence was tested: $s(f_i) = 1 - \frac{KL}{A+B}$. The more frequent a term, the higher $s(f_i)$.

Table 5.8 shows the top 10 words for the SDA corpus for each technique. The large differences between the measures are also illustrated by Table 5.9, showing the number of common words in the top 100 lists for each pair of methods.

[10] For the data sets with less than 10 categories, the χ^2 statistic was only considered valid if none of the A values was zero and if not more than two A-values were less than five.

	df	**discr**	χ^2	E	E'	E''	**KL**	**WKL**
df		38	1	0	1	83	43	73
discr	38		11	0	4	36	68	48
χ^2	1	11		2	1	1	15	2
E	0	0	2		0	0	0	0
E'	1	4	1	0		1	4	1
E''	83	36	1	0	1		43	75
KL	43	68	15	0	4	43		57
WKL	73	48	2	0	1	75	57	

Table 5.9: **Overlap between stopword discrimination measures.** Number of common words among the top 100 of each stopword extraction technique as calculated for the SDA data set.

5.4.2.1 Self-Validation

For each data set and each of the eight measures we calculated "stopwordliness" and sorted the terms for each such scenario in descending order. To gain a first impression of the stopword candidates thus found, they were applied to the document sets from which they had just been derived (i. e. the SPRINGER stopwords were tested on the SPRINGER corpus, the AMAZON stopwords on the AMAZON corpus etc.). While this self-validation procedure does not lead to a valid general evaluation, it serves as an indicator of how well each measure determines the superfluousness of a word in the original context (i. e. with the labels known *a priori*). We used different stoplist lengths, with the cut-off point α set to 50, 100, 250, 500, 1000, 1500, 2000, 2500 and 5000. Figure 5.7 reports on these experiments in a summary fashion by giving just the average results for the nine α values; for the detailed lists consult the Appendix (Table D.13).

The lengths of the vertical lines again indicate the reduction in matrix size caused by each stoplist. Because all morphological, syntactic and semantic information was stripped, one term on the initial stopword candidate list can occasionally eliminate more than one term in the feature set (e. g. if "der"/PRO was found to be a stopword candidate, it is afterwards responsible for eliminating "der"/PRO , "der"/ART , "Der"/NAM, etc.).

On the whole it transpires that the concept of "stopwordliness" is rather difficult to capture, and even with prior knowledge of the correct labels it can be difficult to identify suitable stopwords. For the SPRINGER and SDA sets several methods were "effortlessly" able to identify useful stopwords whereas for the AMAZON and WIKI set it proved very difficult to find lists that improved clustering performance. Therefore we have either failed to find a suitable stopword identification method yet or we must conclude that for certain clustering tasks there are no stopwords (at least not in a significant number).

This being said and kept in mind, we can make a few observations about the individual measures for stopwordliness:

- Document frequency (df) seems to be useful in the very high frequency range. If α (the number of words on the stoplist) is chosen to be in the range of 100, the results are relatively good, but with larger α, performance deteriorates as increasingly more content terms are excluded as well.

	SPRINGER	AMAZON	SDA	WIKI	NZZ	**avg**
df	81.9	79.6	45.8	84.0	70.7	72.4
discr	55.3	66.6	66.8	57.8	69.6	63.2
χ^2	84.3	76.4	109.0	87.2	84.9	88.8
E	56.1	86.1	87.0	82.1	85.1	79.3
E'	107.3	76.2	129.3	90.4	79.8	96.6
E"	49.8	63.2	44.8	67.6	79.3	60.9
KL	55.5	73.2	62.7	57.3	66.6	63.1
WKL	70.3	77.6	41.1	72.4	67.4	65.8

Table 5.10: **Self-validation with rank-sums.** Comparison of different stopword extraction techniques with summed rank-score sums for cluster quality and matrix size reduction. The lower the score, the better.

- Discriminative power (*discr*) shows relatively good results: it considerably reduces the number of non-zero elements in the matrix while leading to consistent results (improving performance for SPRINGER and SDA while not worsening it too much with the others).
- χ^2 and E' both neglect the frequency component and tend to over-emphasise rare terms. The elimination of the latter has comparatively little impact; moreover, there is a big chance that the words thus eliminated would be very important in a different corpus.
- Pure entropy (E) suffers from not taking account of the different category sizes. In cases of uneven distributions of documents over labels, the resulting stoplists contain too many irregularities.
- E'' performs along the lines of *discr*, with roughly similar complexity reductions and cluster quality.
- Relative entropy (KL) takes proper account of the different category sizes and, indirectly, also of frequency.[11] Despite being based on a well-known theoretical concept, its results are only about equal to those of E'' which, using the empirical coefficient p_4, lacks such a foundation.
- The attempt to pay more attention to frequency by weighting KL with document frequency (WKL) does not appear to improve the results. With larger α difficulties similar to those of document frequency (df) start to occur.

All in all it would seem that *discr*, E'' and KL are the best measures for finding stopword *candidates* (see also the summed rank-sum scores[12] in Table 5.10).

The perplexing difficulty of finding suitable stop words for the AMAZON and WIKI corpora *even with a priori knowledge of the labels* has yet to be explained.

5.4.2.2 Cross-Validation

For each data set and each combination of stopwordliness measure and α, new stopword lists were then created from the lists of the four other sets. In a first run the four lists were united; in

[11] Because rare features are less likely to be distributed in accordance with the category sizes than high-frequent ones.

[12] Cf. Section 4.3.3 for definitions.

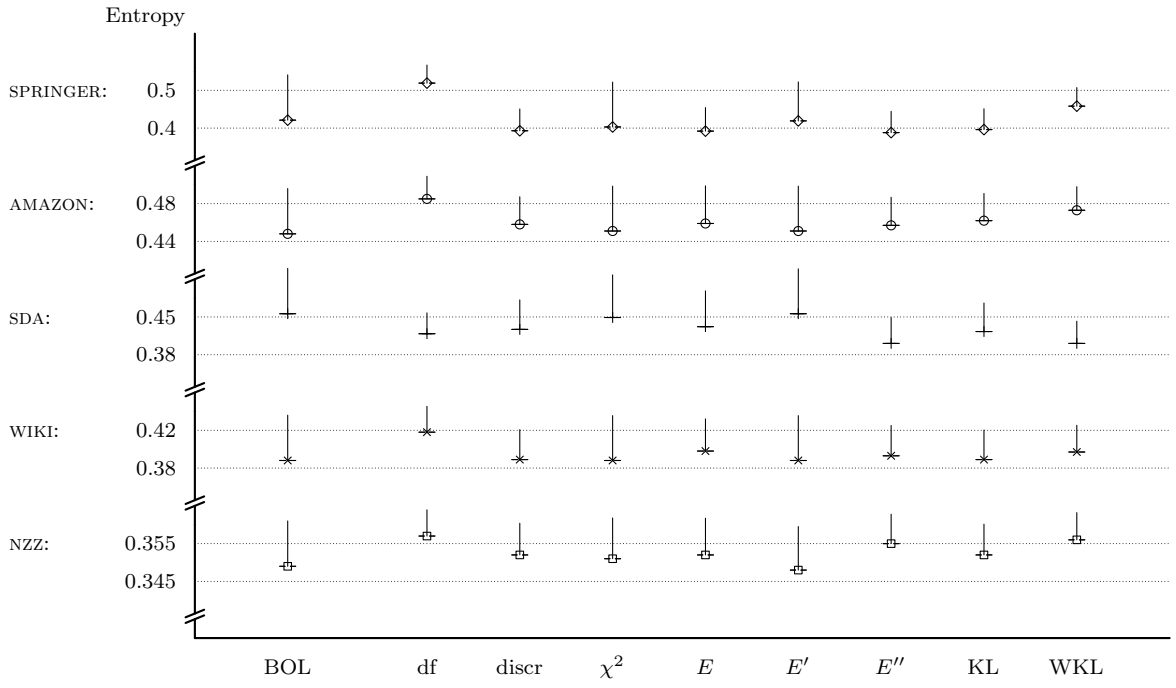


Figure 5.7: **"Self-validation" of different stopword discrimination measures.** Note: The figure only shows the *averages* for the nine different α values! See Tables D.12 and D.13 for the detailed results.

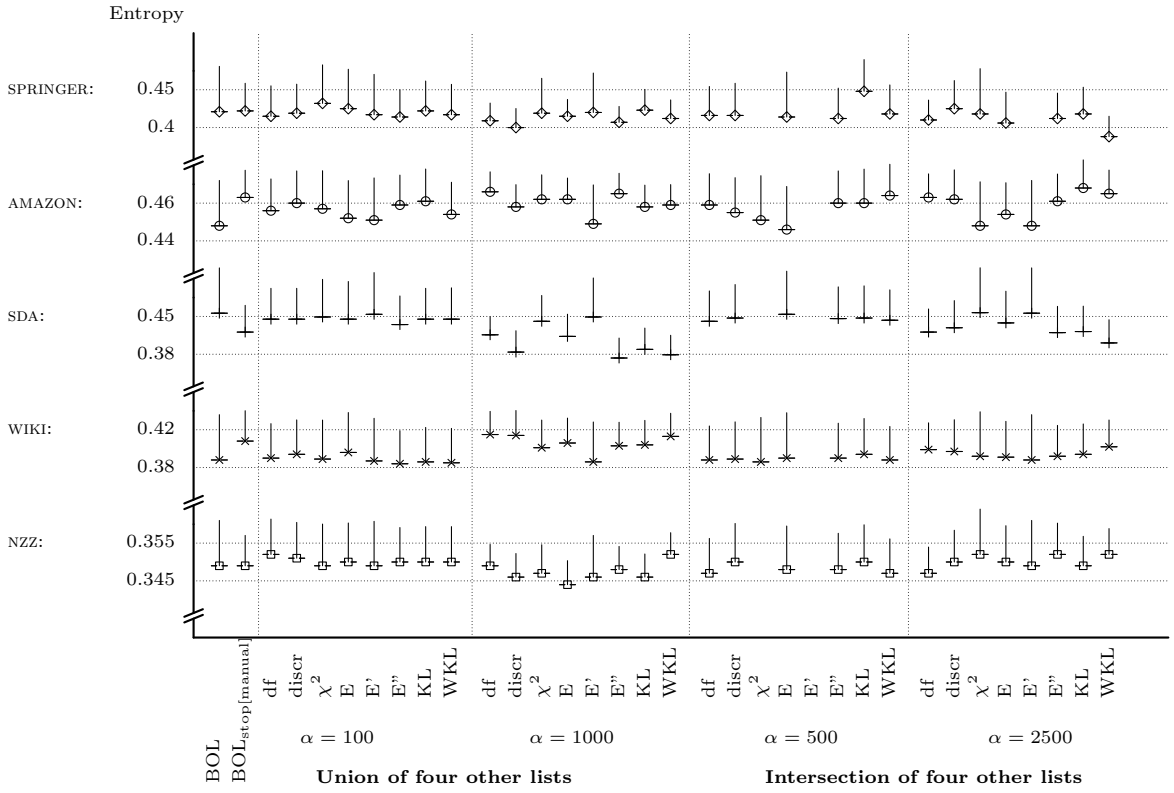


Figure 5.8: **Cross-validation of different stopword discrimination measures.** Results were omitted if the intersection led to an empty stoplist. Only selected values for α (100, 1000 resp. 500, 2500) are shown. For the complete lists of results see Tables D.14 and D.15 in the Appendix.

a second run the intersection of the four lists was used. In both cases the new list was evaluated on the fifth set. Thus, for use on the SPRINGER set, stoplists from AMAZON, SDA, WIKI and NZZ were joined (or intersected), and so on. The results of this cross-validation technique allow us to form a judgement on the usability of automatic stopword extraction methods. The full results are given in the Appendix (Tables D.14 and D.15); a few typical results are rendered in Figure 5.8.

On the whole, the results bear resemblance to those of the previous section. The difficulty of finding a good reduction for the AMAZON set has been confirmed by the fact that just four stoplist combinations led to a (minimal) qualitative improvement. The SDA results were again improved easily by stopword removal, while WIKI could deteriorate if the stoplist was too long and NZZ proved again relatively stable.

The results show that automatic stopword extraction can work satisfactorily, with more than one stopword detection method leading to acceptable results. For a number of individual cases it can even be shown that those lists are superior to the manual list. Using multiple sources with

	SPRINGER	AMAZON	SDA	WIKI	NZZ	**avg**
	Union:					
df	66.6	69.9	66.5	78.7	83.8	73.1
discr	50.1	69.4	56.3	72.6	79.8	65.6
χ^2	109.9	78.6	100.3	74.4	80.2	88.7
E	71.5	74.2	78.8	72.7	63.4	72.1
E'	107.0	83.7	123.9	82.9	88.9	97.3
E"	51.2	72.3	50.8	63.9	61.8	60.0
KL	77.4	77.4	61.7	70.8	68.7	71.2
WKL	63.4	70.7	55.2	76.9	85.6	70.4
manual list	92.5	89.5	79.0	102.0	82.0	89.0
	Intersection:					
df	54.9	60.1	47.7	51.9	51.5	53.2
discr	65.7	64.4	61.1	60.2	70.4	64.3
χ^2	78.5	67.4	107.2	75.2	103.8	86.4
E	51.5	61.1	65.2	77.9	71.4	65.4
E'	93.0	65.5	106.0	83.5	88.0	87.2
E"	46.6	66.7	52.6	57.6	50.4	54.8
KL	71.9	62.7	53.0	73.1	59.4	64.0
WKL	42.6	61.1	40.8	53.3	42.8	48.1
manual list	59.5	64.5	28.0	76.5	49.5	55.6

Table 5.11: **Cross-validation rank-sums for different stopword discrimination techniques.** Rudimentary comparison of different stopword discrimination techniques using summed ranks-scores for cluster quality and matrix size reduction (calculated separately for union and intersection of candidate lists). The smaller the number, the better. The *intersection* rank sums (lower table) are lower because cases with empty stoplists were omitted.

intersection seems to be a reliable procedure for determining a new stoplist.

Comparing the stopword measures among themselves is not trivial because the influence on matrix dimensions, matrix size and cluster quality varies from measurement to measurement. The rank scores described previously are rather crude, but still a way of obtaining an overview: rank sums of clustering entropies and matrix sizes were calculated and for each method these rank sums were again added up and the average taken (all done separately for the union and intersection experiments). The average rank sum (Table 5.11) for each stopword discrimination measure is thus an average value for all the lists arising from $\alpha = 50 \ldots 5000$. For the manual list (last row) just a single measurement exists, of course.

It would seem that on average the best stopword measures are E'' and weighted relative entropy (WKL). The discriminative value is a further viable option but somewhat less good at detecting frequent stopwords. Simple document frequency has also been shown to perform quite well if the different candidate lists are intersected which eliminates some of the greatest drawbacks of document frequency.

Entropy (E), E' and χ^2 are less suitable for stopword detection, mainly because they fail to properly take the quantitative aspect into consideration, leading to the inclusion of too many rare words in the stoplists.

5.4.3 Conclusions

On the whole, we have found stopword removal to be a relatively useful technique for matrix size reduction, with results that actually compare favourably to those achieved by purely statistical reduction methods. However, when compared to the *full* feature set, a decline of the results for the AMAZON and WIKI sets still remained in most cases. We therefore conclude that stopword removal should not be blindly applied for all clustering tasks. For a further discussion of the phenomenon see Section 5.6 below.

It was further demonstrated that several measures exist—E'', (W)KL, *discr*—that capture a word's "stopwordliness" with respect to a given labelled document set with adequacy on par with human judgement. These measures have been successfully used to extract new stoplists by combining the lists of different data sets. The results are encouraging as the best measures tend to perform somewhat better than the manually created stoplist.

5.5 Part-of-Speech Selection

In this section we discuss matrix reduction based on the part-of-speech tags of the individual tokens (see Section 5.2.1 for the details of POS tagging). Documents are thereby reduced to words belonging to specific POS categories.

The results (Figure 5.9) prompt a number of observations:

- Nouns (SUB) are not only the most numerous features but also strictly essential for clustering.
- Adjectives (ADJ) are the second most useful ingredients. Their addition improves results in all five cases. [see column 2 in the figure]
- Names (NAM and NAM_{all}) come next. Their addition to nouns helps in four of five cases. [7,8] However, their addition to nouns and adjectives only helps in two cases—casting a certain doubt on their usefulness. [12]
- Adding verbs, appositions, numbers or unknown tokens [3–6] is less helpful. Positive and negative effects occur with about equal frequency.
- Verbs, even though an open word class, seem often to have too general a meaning [3,15,17,18], i. e. they function more or less as stop words. However, the results remain unconvincing even with the exclusion of modal verbs.[13]
- Based on the above data it is difficult to give a general recommendation. *Ceteris paribus* it would seem that either SUB/ADJ [2] or SUB/ADJ/NAM_{all} [12] offer the best results. In four of five cases, both these combinations lead to qualitative improvements over ordinary stopword removal [20] (with the additional benefit of a heavier matrix size reduction).

Feature selection based on POS tags is thus a viable alternative to stopword removal. The general tendency is the same as for the previous sections: "difficulties" with AMAZON and WIKI, relatively straightforward improvements for SPRINGER and SDA.

[13] Removing modal verbs (können, dürfen, mögen/möchten, müssen, sollen and wollen) from a SUB/VRB representation improved the results only slightly (with one negative exception):

	SPRINGER	AMAZON	SDA	WIKI	NZZ
SUB, VRB_{all}	0.448[0.018]	0.467[0.008]	0.486[0.001]	0.403[0.011]	0.389[0.024]
$\text{SUB, VRB}_{\text{no modals}}$	0.455[0.018]	0.464[0.007]	0.485[0.001]	0.399[0.016]	0.378[0.020]

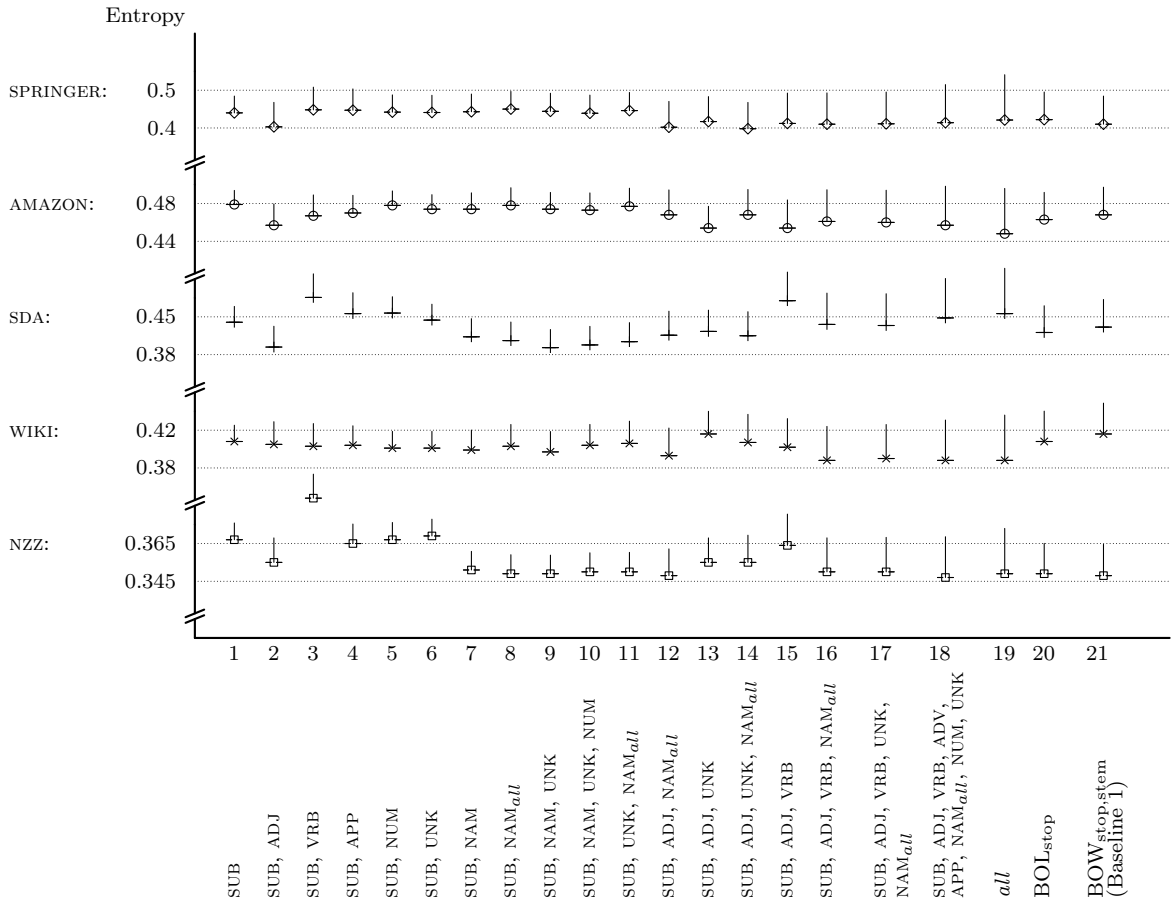


Figure 5.9: **POS-based feature selection.** Vertical lines correlate with the number of features surviving the selection step. (For the underlying numbers refer to Table D.18.)

5.6 Comparison of Matrix Reduction Techniques

The previous sections discussed several matrix size reduction techniques. We found that substantial reductions in matrix size of up to 50% or more were sometimes possible without negatively affecting clustering results. It was also shown that in terms of clustering quality, the data sets behaved quite differently and that even the universally approved stopword removal technique can, in terms of cluster quality, have negative effects.

Of the methods discussed, we found that the more sophisticated methods such as stopword removal with extracted stoplists and especially POS selection (SUB/ADJ) tended to lead to a better cluster quality than the statistical cluster pruning techniques (when measured at a comparable reduction factor).

Throughout these experiments, we were rather surprised to find that for AMAZON and WIKI none of the matrix reduction techniques managed to improve cluster *quality* (i. e. benefits only occurred in terms of matrix size). We hypothesize that this phenomenon is correlated to the number of clusters to be built and that the larger the number of clusters/labels (21 resp. 22 for AMAZON and WIKI, as opposed to 5 resp. 7 for the other three sets) the more detail is needed and the more sensitive are the results with regard to feature reduction.[14]

In order to test this hypothesis, we repeated the manual stopword experiment for a number of subsets from AMAZON and WIKI. For each value of $k = 5, 7 \ldots 17, 19$ we selected at random five subsets of k labels and clustered the documents of those subsets separately. For each subset we then compared the results prior and after stopword removal, with results as in Table 5.12. They confirm our hypothesis that stopword removal gradually gains in effectiveness with a shrinking number of clusters.

Furthermore, it is generally notable how little the NZZ results are influenced by most techniques (keep in mind that in the various figures the choice of scale leads to a stronger magnification of the NZZ effects than with the other sets). We believe that the large amount of redundancy (as manifested by the big average length) may be accountable for this phenomenon. In other words, unlike with short texts, the large amount of content evidence drowns the "stop word noise". A brief experiment was conducted in which the NZZ documents were mutilated after 20, 50, 100, 150, 200, 250, 300, 400, 500 tokens. The results prior and after application of the manual stoplist (Figure 5.13) indeed showed a rising sensitivity towards the presence/absence of stopwords when documents are mutilated.

It can therefore be concluded that stopword removal is more likely to increase clustering quality ...

- with a small number of clusters than with a large one,
- with short documents than with long ones.

[14] Unlike the SPRINGER, SDA and NZZ texts, the AMAZON and WIKI texts are also written by a much more diverse authorship and with far less general stylistic or orthographic conventions than is the case with the former three. However, it appears difficult to explain how this could lead to such strong divergences as have been repeatedly observed.

number of clusters/labels		stopword removal harmful	stopword removal helpful	no effect
$k = 5$	AMAZON	0	**5**	
	WIKI	1	**3**	1
$k = 7$	AMAZON	0	**5**	
	WIKI	2	**3**	
$k = 9$	AMAZON	1	**4**	
	WIKI	0	**5**	
$k = 11$	AMAZON	1	**3**	1
	WIKI	1	**4**	
$k = 13$	AMAZON	1	**4**	
	WIKI	2	**3**	
$k = 15$	AMAZON	**3**	1	1
	WIKI	2	**3**	
$k = 17$	AMAZON	**3**	2	
	WIKI	2	**3**	
$k = 19$	AMAZON	**3**	2	
	WIKI	2	**3**	
all ($k = 21/22$)	AMAZON	**1**	0	
	WIKI	**1**	0	

Table 5.12: **Reduced number of clusters.** Experiments on AMAZON and WIKI with just a *subset* of the document collections and a reduced number of clusters. For each value of k five subsets were chosen and the results prior and after stopword removal (with BOL) compared. On each line the position of the bold number indicates whether stopword removal was on average helpful or harmful. The individual results are given in the Appendix (Tables D.16 and D.17).

	20	50	100	150	200	250	300	400	500	$n \to \infty$
BOL	0.750	0.522	0.446	0.428	0.415	0.407	0.401	0.387	0.376	0.349
BOL_{stop}	0.727	0.500	0.441	0.426	0.414	0.407	0.402	0.387	0.378	0.349

Table 5.13: **Text mutilation.** Stopword removal with NZZ texts mutilated after the first n tokens.

5.7 Feature Weighting Techniques

Although it is rarely done, the reduction techniques discussed in the previous four sections can be regarded as a special, binary case of feature weighting. If our main aim is not so much to reduce matrix size but to improve clustering quality, it should naturally be asked whether or not a continuous feature weighting approach might not be preferable to this rigid 0/1 method.

In the present section we briefly discuss three techniques that aim to increase the weights of "important" terms without altogether discarding the "unimportant" ones.

5.7.1 Part-of-Speech Weighting

In the first experiment, certain POS categories were given extra weights vis-à-vis the rest (Figure 5.10). Based on the conclusions of Section 5.5 we concentrated on nouns, adjectives and proper names. In accordance with the earlier experiments we find that SPRINGER and SDA benefit from the extra weighting of nouns, adjectives and, to a lesser degree, names, whereas the AMAZON and WIKI sets often show an increase in entropy. It is notable, however, that compared to the earlier experiments the NZZ set appears to profit from most the POS weighting method most. Though probably not statistically significant, AMAZON is also improved minimally if adjectives and nouns are given some extra weight.

NAM-tokens seem generally less important than one would intuitively think.

Nevertheless, judging on the evidence it appears that, even without further knowledge about the data set, giving moderate extra weight to nouns and adjectives is a relatively safe way of attempting to boost clustering quality without taking the risk of eliminating important terms.

5.7.2 Stopword Weighting

From what has been just said, it is natural to ask whether such a smoothing approach cannot be applied to stopwords as well. Rather than removing them altogether, we can try to decrease their weights. Figure 5.11 indeed shows some improvements if the stopwords are weighted with a smooth factor $0 < \gamma < 1$, but the effects are insignificant and can also occur in the other direction.

Down-weighting stopwords might make sense if no information about clustering behaviour with regard to stopwords is known. But otherwise (i. e. if we already know if the reaction tends to be positive as with SDAor negative as with AMAZON or WIKI) a strict 0/1 decision is preferable, in particular since down-weighting instead of removing does not come with a reduction in matrix size.

5.7.3 Weighting Front Nouns

Table D.21 reports on a related experiment. Since a large number of texts start by mentioning some of the very key concepts in the first one or two sentences, we gave extra weights to the first five or ten nouns in each text. The results are rather sobering; a small insignificant improvement could only be observed with the NZZ set. Perhaps such an approach is only to be considered with texts of sufficient length.

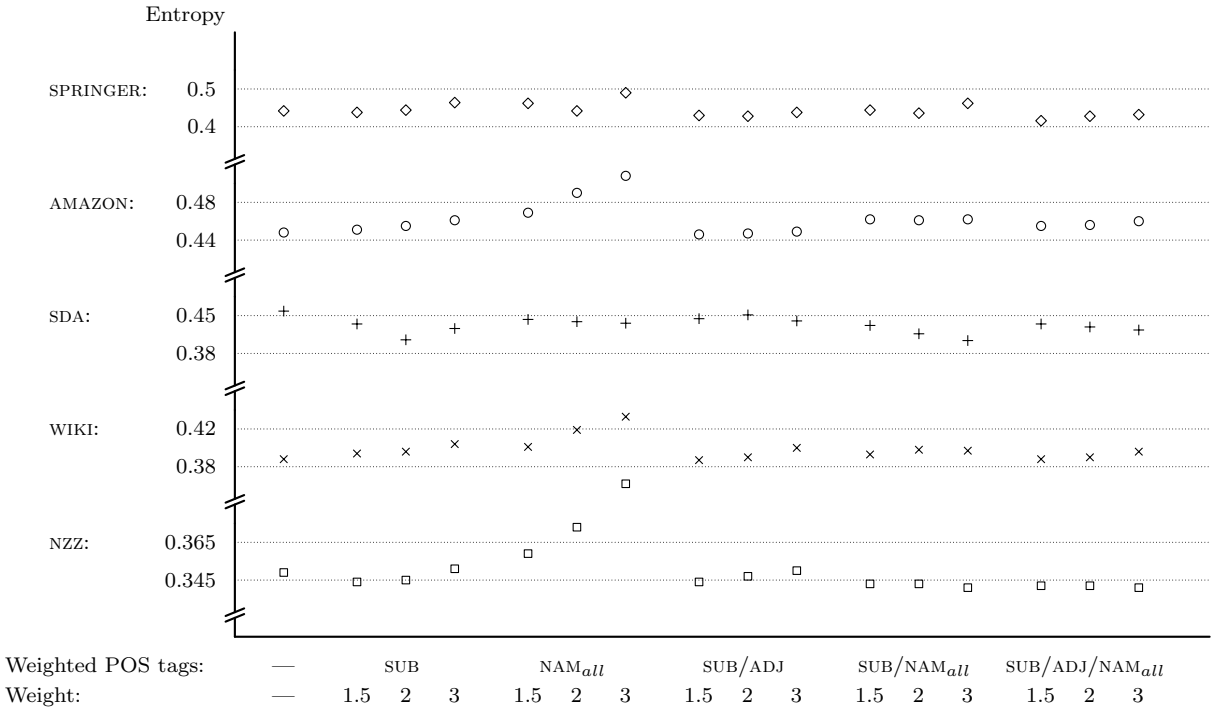


Figure 5.10: **Lending extra weight to chosen POS categories.** (For the underlying numbers refer to Table D.19.)

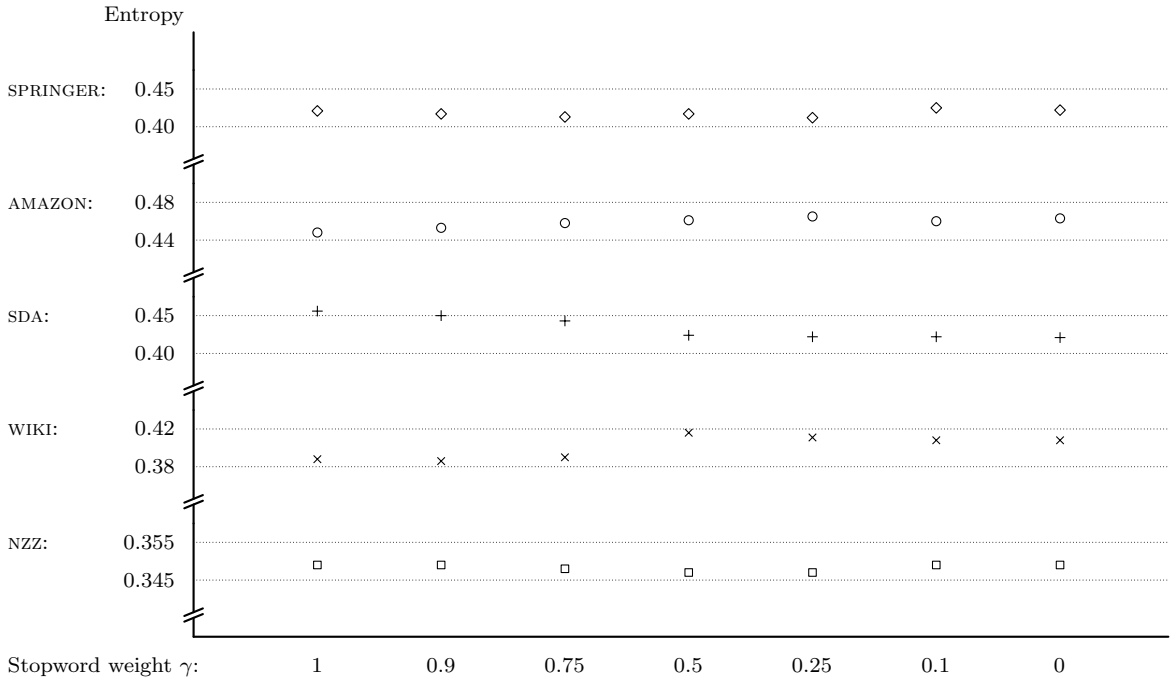


Figure 5.11: **Stopwords weighting instead of elimination.** From treating stopwords like all other words ($\gamma = 1$) to complete stopword removal ($\gamma = 0$). (For the underlying numbers refer to Table D.20.)

5.8 Summary of Reduction Experiments and NLP

Let us try to summarise the impact of NLP methods on the representation reduction techniques discussed in this chapter. In Figure 5.12 we have selected of the flood of experimental data what appeared to be the most significant numbers for various methods. The results achieved by purely "statistical" methods are numbered S1–S5, those making use of linguistic knowledge L1–L6. M1–M2 show some results by "statistical" methods but on a linguistic base (lemmata).

A simple comparison shows that the bag-of-lemmata (L1) leads to better results than the bag-of-words (S1). However, with stemming (S2) the BOW results become just as good as those from lemmatising.

It has been demonstrated that the creation of stoplists is amenable to statistical methods with results that compete with a manually created list. Our stopword discrimination experiments were all conducted with lemmata (M1), but there is no reason to doubt that the same procedure would also work with the bag-of-words. Stopword removal has comparable effects on the bag-of-words (S3, S4) and the bag-of-lemmata (L2).

If we compare the aggressive statistical matrix reduction approach (stopping and stemming; S4) with a linguistic alternative (lemmatising and restriction to nouns, names and adjectives; L3), the latter scores better. The comparison between statistical pruning of lemmata (M2) and POS selection (L3–L5) also turns out in favour of the linguistic variant(s).

Finally, the POS weighting approach (L4, L5) leads to results that are more or less equal to the *best* of S1–S4, while beating the individual choices more or less clearly.

The comparison of the "best each" lines (S5, L6) further confirms that the linguistic methods bear potential to improve clustering quality beyond the possibilities of statistical methods. At the same time it must be noted that the evidence is not overwhelming, the outcome of the experiments rarely reaching a level of firm statistical significance. We have also observed that the effects of representation methods were not the same for all data sets. What proved useful in one instance, could prove disadvantageous in another.

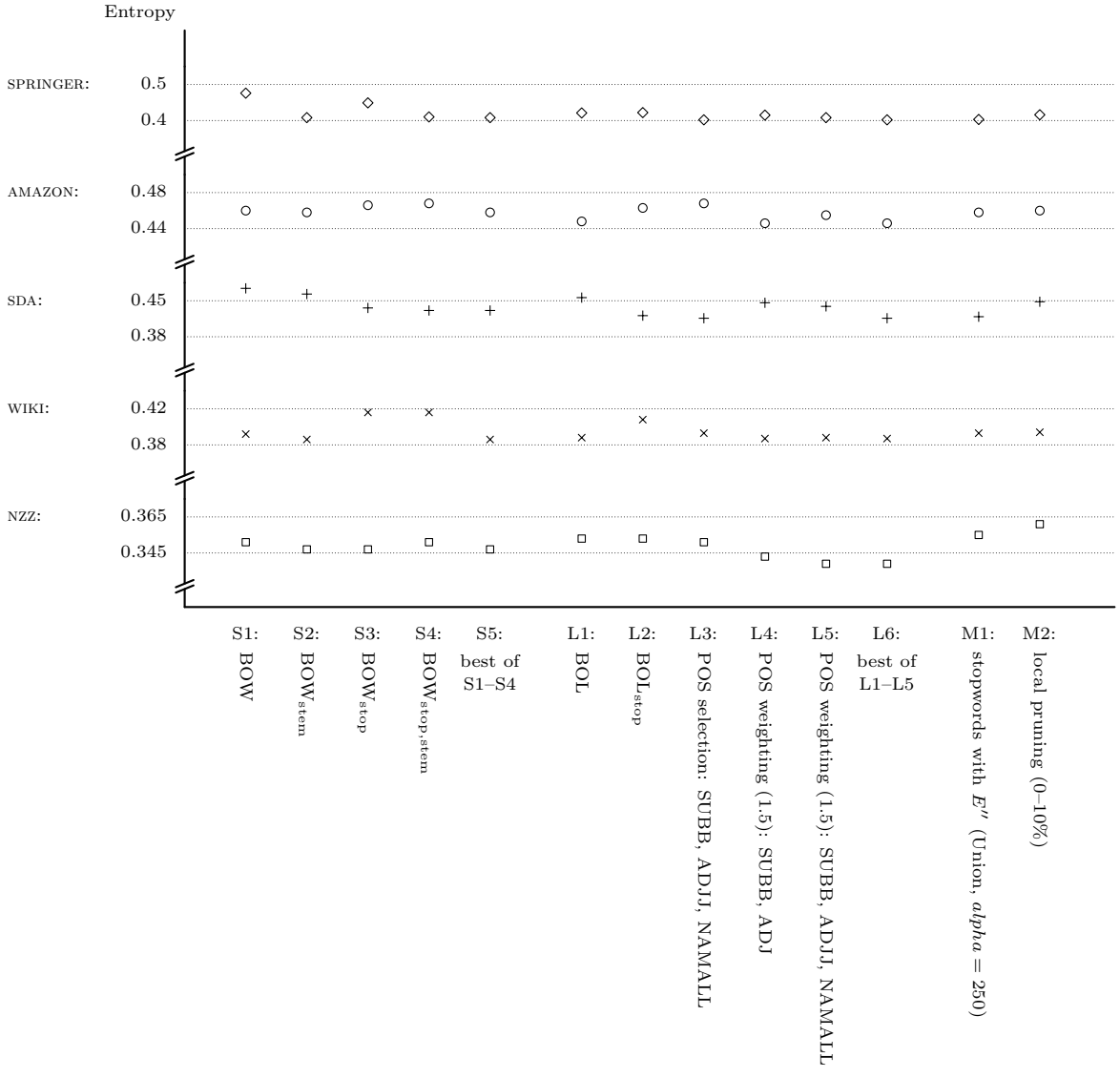


Figure 5.12: **Selected linguistic and statistical reduction methods in comparison.** (For the underlying numbers refer to Table D.22.)

Chapter 6

Enhanced Document Representations Using Natural Language Processing

Say the word and you'll be free
Say the word and be like me
Say the word I'm thinking of
Have you heard the word is love?
It's so fine, It's sunshine
It's the word, love
In the beginning I misunderstood
But now I've got it, the word is good.

The Beatles (*"The Word"*, 1965)

The word may be good, as in the Beatles song, but sometimes it may not be good enough. In the present chapter we shall try to look beyond the individual word in order to also capture some of its meaning and context. We investigate how and to what extent information on a morphological, syntactic and semantic level can help us to improve document representation and hence clustering results.

On the *morphological* level, we try to analyse the words and extract some additional information from the way they are built (Section 6.1). On the *syntactic* level we look at the words in their immediate context within a sentence (Section 6.2) while on the *semantic* level we try to abstract from the outward word forms and penetrate to the actual meaning of each word (Section 6.3).

It is clear that these techniques are not at all aimed at optimising speed or reducing matrix size. Their sole purpose is to improve clustering *quality*.

6.1 Using Morphological Information

Morphological analysis has already been used at an earlier stage, when word forms were mapped to their lemmata (Section 5.2). However, a detailed analysis of each word can also be used for

further purposes. Especially in languages with a strong tendency towards compounding it can be useful to split compound terms into their constituents (cf. Section 3.4.2.3).

The present section reports on a number of experiments in which the compounds in each data set were analysed and split up according to different rules. All are based on the morphological analyses provided by GERTWOL. In order to keep the conditions equal, stopword removal was performed for all sets prior to compound splitting, despite the previous observation that stopword removal may not be good in all cases. As we are mainly interested in the relative effect of compound splitting, this was considered an acceptable simplification (but see Section 6.1.4 below for a further discussion of this aspect).

6.1.1 Mechanical Compounds

First we look at features which can be split into constituents without a real morphological analysis, i. e. features such as "Rechtschreibe-Kommission" and "links/rechts". We call such features of the type A-B or A/B "mechanical compounds". The impact of splitting "mechanical" compounds (which are treated as a unit by GERTWOL) appears to be rather small, both with the BOW and BOL models (Figure 6.1). They can easily be recognised and split up without further ado—in fact, it is what a number of tokenisers do automatically.

As a rule we retained both the parts and their original compounds as features. The few experiments where the original compounds were dismissed after splitting are marked with a star (*).

6.1.2 Organic Compounds

Let us now turn to real compounds which are not discernible as such on the symbolic level but require a true morphological analysis. We call them "organic" compounds. GERTWOL provides the necessary information by indicating strong compound-boundaries with the symbol "#". For instance, the term "Freistaat" (free state) is analysed thus:

"Freistaat" → "Frei#staat",

giving us the two constituents "frei" and "staat".

Often, however, compound resolution is less straightforward. For instance, some words consist of more than two constituents or allow several different segmentations. An example is "Fussballerleben", for which GERTWOL offers three principal analyses:[1]

Fussballerleben →
1) Fuß#ball#er|leb∼en (foot—ball—experience),
2) Fuß#baller#leb∼en (foot—bang! {"ballern"}—life),
3) Fuß|ball∼er#leb∼en (football player—life).

As before, we follow Volk's heuristic to choose the most likely lemma (Volk, 1999). Here it favours interpretation 3) which is indeed the most likely usage of the term. This method may not always lead to the correct segmentation (indeed, there are cases where even a human being will be unable to predict the correct or most likely segmentation without knowledge of the context), but it is fairly accurate and leads to consistent solutions.

Our ultimate aim in compound splitting is to strengthen those individual constituents of a compound that also occur at least once somewhere else in the corpus, either freely or as part of a different compound. Not all compounds behave as neatly as "Freistaat" which produces

[1] Note that ∼ indicates suffixes, while | points to "weak" compound boundaries (which are of little interest to us).

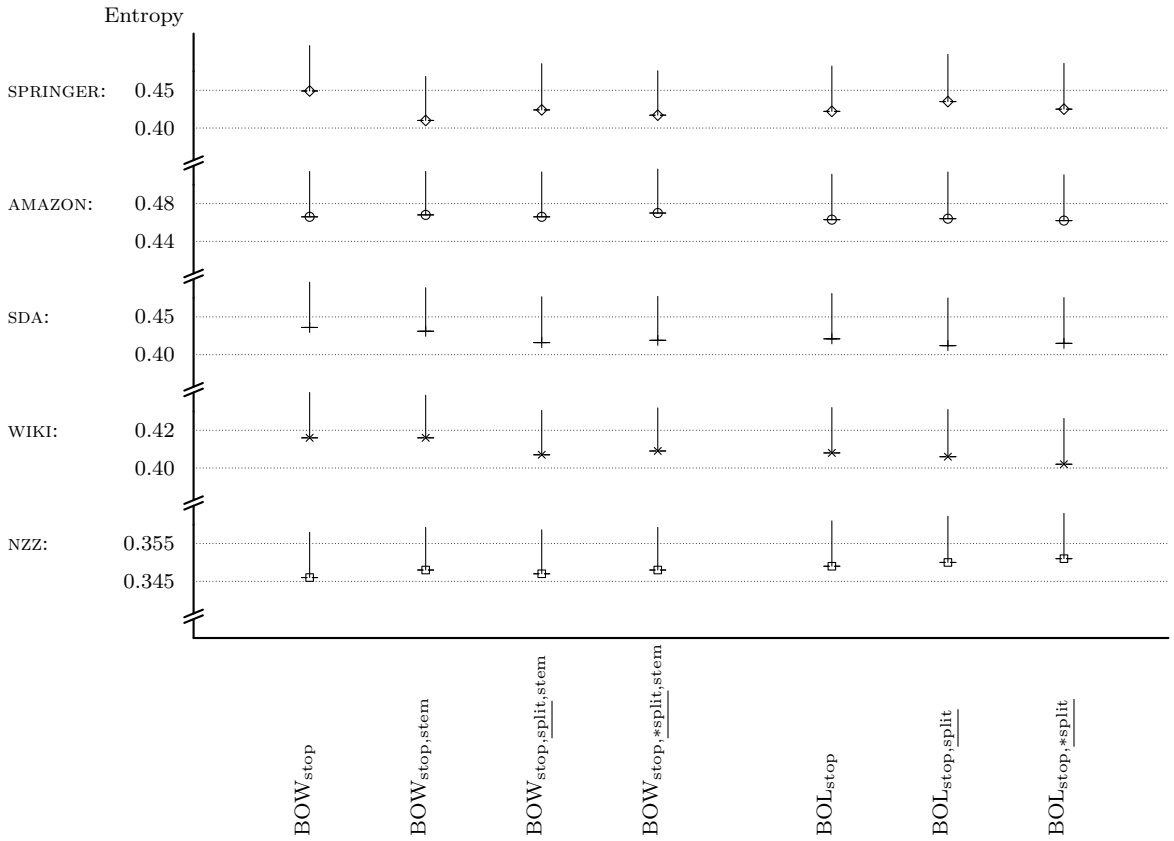


Figure 6.1: **Splitting mechanical compounds.** BOW and BOL extended by splitting "mechanical" compounds (A-B and A/B features). Stars ("*split") refer to experiments in which the compound token was *discarded* after analysis and replacement by the constituents. (For the underlying numbers refer to Table D.23.)

two perfectly ordinary lemmata, ready for immediate use as features ("frei", "staat"). In fact, compound *building* often involves some sort of inflection from the partaking lemmata, so some sort of standardisation becomes necessary when moving in the opposite direction. Usually the last element in a compound behaves regularly and inflection occurs with the earlier, modifying constituent(s).

The necessary standardisation can be achieved in two ways: either by simple stemming (which, of course, must then be applied to all features, not just the constituents), or by further analysis. In accordance with the linguistic approach followed in this work, we opted for the latter solution, which meant that using GERTWOL for a second time, we tried to reduce all compound parts to proper lemmata.

Four possible cases can be distinguished:

- "**Frei**staat"—the modifying part ("frei") is already in lemmatised form and is therefore easily recognised by GERTWOL.
- "**Hund**eleine"—extra material appears between the parts, "gluing" them together. Fortunately, GERTWOL analyses these so-called *fugenlaute* (such as -e-, -s-, -en-, -n-) properly and offers "Hund\e#leine", where the backslash symbol allows us to identify and remove the extra -e-, retrieving "hund" and "leine", for which GERTWOL will provide suitable POS tags.
- "**Fahr**schule"—the opposite case is also possible: a lemma is abbreviated, usually to its stem. Although these stems may occur only rarely on their own in a normal text, if fed to GERTWOL, they are often recognised as special (imperative) verb forms. Thus, when we analyse "fahr" separately, GERTWOL will (correctly) retrieve the lemma "fahren/VRB".
- "**Adreß**etikett"—in some other cases GERTWOL will not know what to do with a nominal stem such as "adreß" in "Adreß#etikette" (or "geschicht" in "Geschicht\s#schreib∼ung"). In order to still retrieve an ordinary lemma, we feed GERTWOL these stems with the extra suffixes -e, -en and -er appended. At least one of them usually allows GERTWOL to come up with a plausible interpretation.

This procedure allows us to effectively split almost all compound terms in our corpora into sensible lemmata. One anomaly must be mentioned: there is a relatively large number of word stems which can generate both nouns and verbs. Unless those word stems are nouns themselves, they are always analysed as verbs (imperatives) even though that might not be intuitively correct. For instance, we would prefer "Schul#haus" to be reduced to "Schule/SUB" and "Haus/SUB", but what we obtain is "schulen/VRB" and "Haus/SUB". Using word statistics it would be possible to distinguish cases where a noun interpretation (e. g. by adding the suffix -e) is more likely than the verbal interpretation, but such a distinction was omitted here.

Compounding is a major facet of the German language and so we should expect compound resolution to have a noticeable impact on our results. Figure 6.2 confirms this expectation. The positive impact on SPRINGER and SDA is comparatively huge, and the effect on the other three data sets is still quite remarkable after seeing how "difficult" they had been in the previous chapter.

As to the question whether or not the initial compounds should be retained or whether it is advisable to drop them after replacing them by the constituents, we observe that keeping the compounds makes a difference and that the speed gains obtained by dropping the original compounds must be paid for by a decidedly lower quality of the clustering results. Our experiments thus lead us to a different conclusion than Rosell (2003), who discards the compound terms after splitting.

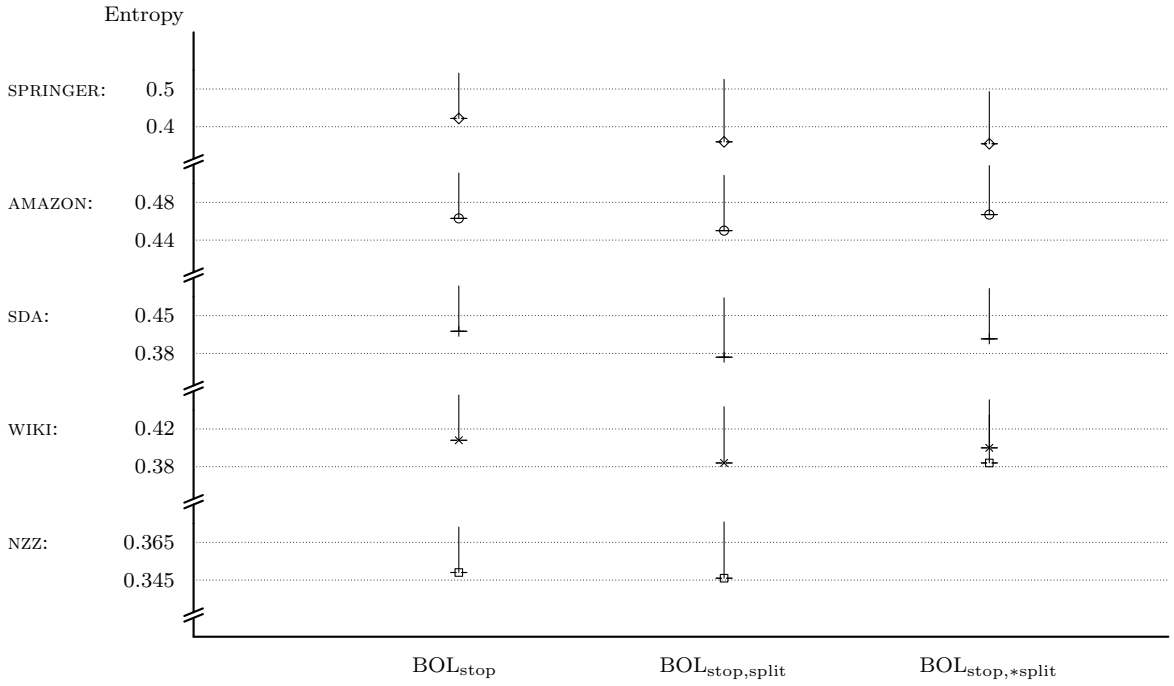


Figure 6.2: **Splitting all compounds** (after stopword removal). The third column refers to experiments in which the compounds were discarded after splitting (which had a very strong negative impact on the NZZ result). (For the underlying numbers refer to Table D.24.)

	SPRINGER	AMAZON	SDA	WIKI	NZZ
types (overall)	35,087	440,883	317,783	618,225	757,072
pseudo-compounds	1,349	9,840	5,357	10,805	15,492
mechanical compounds (non-pseudo)	2,001	45,329	51,373	90,093	74,479
purely organic compounds	13,997	126,153	116,112	227,374	256,380
all organic compounds	14,562	134,237	130,951	244,430	273,789
all compounds	17,313	181,474	172,651	328,395	345,074
percentage of organic compounds	41.5%	30.5%	41.2%	32.3%	44.3%
percentage of all compounds	49.4%	41.2%	54.3%	43.4%	55.8%

Table 6.1: **Compound statistics** (after stopword removal). Pseudo-compounds start or end with a hyphen. Mechanical compounds have an embedded hyphen or slash. Purely organic compounds have neither slash nor hyphen but word boundaries detected by GERTWOL.

6.1.3 Restrictive Compound Splitting

The previous section has shown that compound resolution is in most cases a very effective means of improving document representation and clustering results. Nevertheless, the question arises whether the approach can be modified so as to improve results further, and in particular to improve the results of those data sets that showed only moderate improvements.

In German, compounds are very frequent. Table 6.1 provides an overview of the different compound types occurring in our corpora. Terms beginning or ending with a hyphen are shown separately as "pseudo-compounds". Many such pseudo-compounds of the "-X" type are artifacts, resulting from non-conform typesetting (and left unaltered in previous processing steps). In our corpora the number of "real" (i. e. non-pseudo), organic compounds amounts to 30–45% of all types, i. e. a very substantial part of the whole vocabulary.

Table 6.2 shows organic compounds and their POS categories. Actual compounds occur in the ADJ, SUB and NAM classes (the latter mostly as geographical terms, e. g. "Nord#atlantik). The rest are almost without exception spurious cases. Verbal compounds are usually either groups such as "gesagt/geschrieben", English terms ("Real-Time"), misspellings or mis-tagged terms. The same applies to the few cases listed under ART, APP, KON, PTK, PRO and PUN. UNK usually refers to first parts in constructions such as "Hals- und Beinbruch" whose POS is not yet further identified by our initial tagging/lemmatising procedure. Compound adverbs are usually of the numeric sort ("sieben#hundertsten#mal"—seven hundredth time) or foreign language names with a wrong tag. NAM_1 are usually hyphenated first name combinations ("Hans-Peter") and NAM_2 hyphenated combinations of second names ("Rimsky-Korsakov").

This being said, looking at the large ADJ and SUB classes, we find that indeed most of their members are proper compounds and worthwhile to be considered for splitting (unlike the exceptions in the other POS classes).

Given the large number of compounds, it might pay off to examine a compound more closely before splitting it. Indeed, a compound that is used very often over a prolonged time tends to assume an independent meaning of its own and it loses its transparency (e. g. "Nieder#lage"). Other compounds (so-called "bahuvrihis") have a meaning that is not at all contained in any of the constituents (e. g. "Grün#schnabel"—greenhorn). In these cases it is obviously undesirable to split the compounds.

Deciding which compounds to split and which to keep is a non-trivial task (cf. Matthiesen, 1999). In principle, we want to avoid splitting compounds that are *lexicalised* (have become

	SPRINGER	AMAZON	SDA	WIKI	NZZ
ADJ	1,624	14,203	7,689	21,206	22,199
ADV	26	311	161	451	468
APP	11	27	25	34	27
ART		1			
KON	1	1	1	2	1
NAM	162	5,898	3,422	14,429	5,832
NAM_1	16	921	447	1,262	869
NAM_2	7	326	100	390	308
NUM	7	317	182	1,427	541
PRO		2	1	13	3
PTK		1	3	8	1
PUN		1		1	
SUB	14,164	148,877	155,390	277,519	299,956
UNK	214	1,094	928	1,471	2,690
VRB	6	718	105	817	228

Table 6.2: **"Organic" compounds and their POS tags (types).**

"common words") since their meaning is often no longer associated with the constituents.

Using various properties of compounds, a wide number of criteria can be imagined to select a subset or apply restrictions on the constituents. We subjected the following plausible approaches to a further test:

l *number*: Require compounds to exceed a certain *length* threshold. Idea: longer strings are more likely to owe their existence to "ad-hoc" compounding than short strings, and therefore it makes more sense to split them up. Short compounds are more likely to be frequent constructions, with an according tendency towards lexicalisation.

d *number*: Put a maximum *document frequency* limit on compounds. Idea: Compounds occurring with a certain frequency in the corpus are good features in themselves, and do not need to be split up. Furthermore, frequent features also tend to be lexicalised features. Rare features, on the other hand, are more likely to have increased usefulness if split up.

c $\{AN\}$: Apply POS restrictions on the compounds, splitting only compounds belonging to certain POS classes (A=ADJ, N=SUB). Idea: not all word types are equally likely to contain interesting information. Note that NAM, NAM_1 and NAM_2 features were exempted from splitting in all experiments, even without any restrictions.

sS $\{ANV\}$: Apply similar POS restrictions on the *final constituent* of a compound; discard that constituent unless it belongs to the desired POS group (A=ADJ, N=SUB, V=VRB).

sM $\{ANV\}$: Apply POS restriction on the *leading constituents* of a compound (all but the last).

r $\{1{,}000|10{,}000\}$: Restrict compound splitting to compounds that do not occur in a standard lexicon. Simple lexica consisting of the 1,000 and 10,000 most frequent German terms were used, found in the "Wortschatz-Lexikon" of Leipsic University.[2]

[2] `http://wortschatz.uni-leipzig.de/html/wliste.html`

rC: Restrict constituents to words that occur freely in the corpus. This approach keeps the additional complexity within limits and avoids splitting compounds whose constituents are not likely to be relevant in the context of that corpus anyway.

Finally, with compounds of three or more parts the question arises whether or not they should be split at each boundary. For example, it might make more sense to divide "Handballtorwart" (handball-goalkeeper) just into "Handball" and "Torwart" rather than "Hand", "Ball", "Tor" and "Wart". The splitting points can be determined in various ways; we tested two approaches:

x: Consider *all* single and all multi-part constituents (i. e. Handballtorwart → [Hand, Handball, Handballtor, Ball, Balltor, Balltorwart, Tor, Torwart, Wart]). Naturally, not all features thus found have a real meaning. E. g. "Balltor" is unlikely ever to enter a German dictionary. Nevertheless, for clustering it is unlikely to do any harm and may under certain circumstances even be useful.

X: Consider the longest multi-part constituents at the beginning and at the end of each compound (e. g. Handballtorwart → [Handballtor, Balltorwart]). Typically, we would want to combine this method with a restriction on constituents that also occur independently in the corpus (**rC**). Then the analysis is likely to become: Handballtorwart → [Handballtor, Torwart] (which definitely makes sense).

Figure 6.3 reports on the results of these compound refinement experiments. The benchmark is the second column, i. e. the results gained by simply splitting all compounds to their smallest parts ("split all"). It appears that systematic improvements are very difficult to achieve. Neither minimum length nor POS restrictions lead to improvements. Nor do lexical restrictions help. Even special treatment for multi-part sub-compounds does not generally make much difference (an exception being the XrC variant on WIKI). The only promising path seems to be setting upper limits for the document frequencies of each compound, omitting to split the most frequent ones. But just where to set this maximum document frequency limit seems to be impossible to decide. For the SPRINGER and NZZ sets low numbers seem to be best, while AMAZON, SDA and WIKI fare better with an upper limit in the 100–200 band. Both in absolute and in relative frequencies it is therefore not possible to draw more precise conclusions than that limiting document frequency is the most promising restriction method for compound splitting.

6.1.4 Conclusions

In a productive language such as German, making use of morphological information by means of splitting compounds has been shown to improve clustering results. Even here, however, differences among the individual data sets could be observed and for AMAZON and WIKI the effects were once more relatively small and often just balancing the loss in clustering accuracy that was incurred by stopword removal.

Various refinement techniques have been tested for compound splitting. Most of them have failed to achieve the aim, only skipping (very) frequent compounds has shown some promise.

In order to have comparable test situations for all data sets, our investigations have been based on the BOL_{stop} model. However, since stopword removal had been shown to be rather harmful for the AMAZON and WIKI sets, we repeated some of the compound splitting experiments for these two sets, but this time with stopwords retained. Surprisingly, the resulting scores turned out to be quite similar (in absolute terms) to those with stopword removal. In other words, if stopwords were retained in the document representation, compound splitting did not achieve much. If stopwords were omitted, compound splitting was more successful, but not really beyond the point which had already been achieved by pure lemmatising.

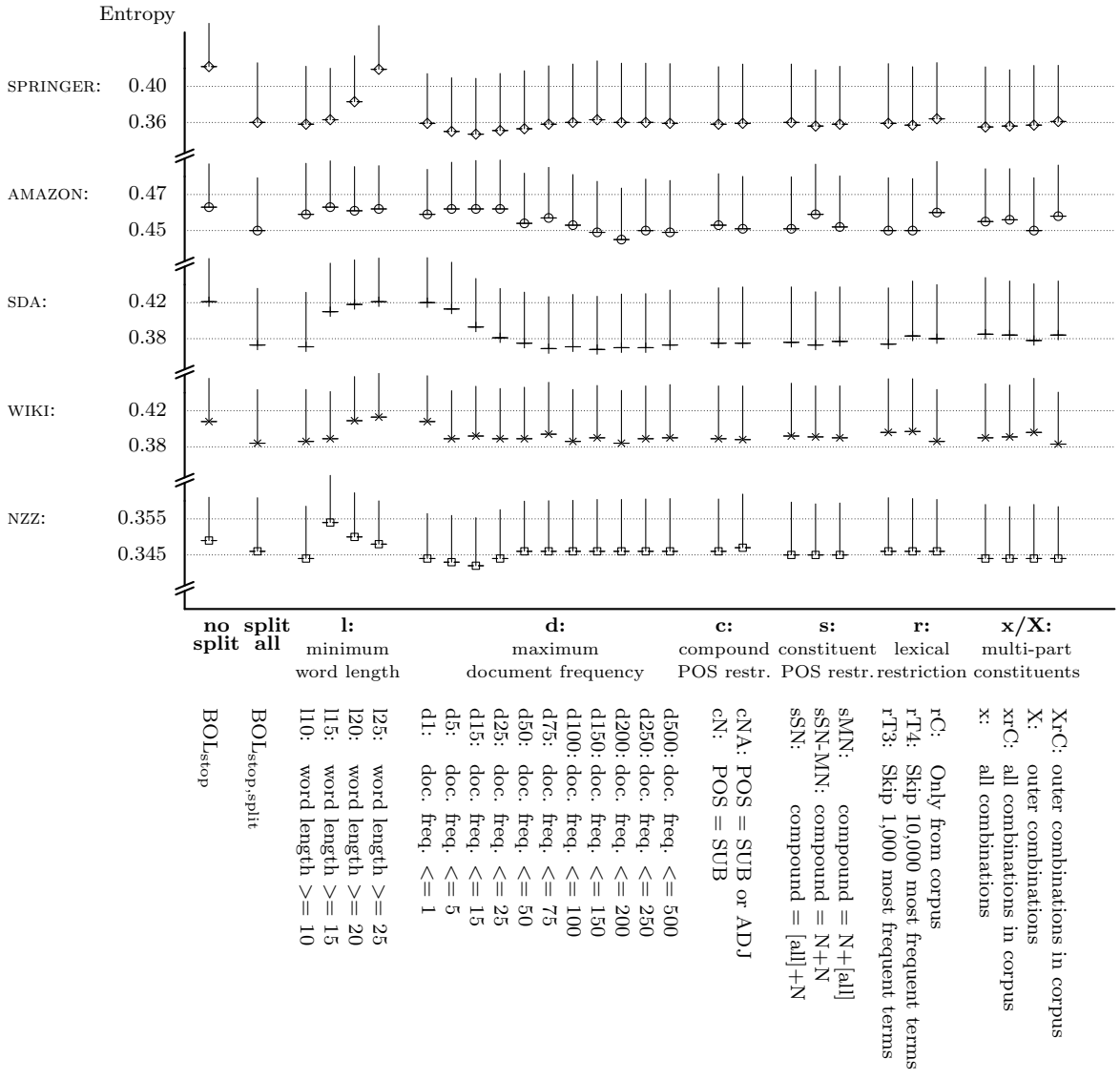


Figure 6.3: **Modifications to compound splitting.** On the whole the baseline $\text{BOL}_{\text{stop, split}}$ (second column) was difficult to beat. (For the underlying numbers refer to Table D.25.)

When the same experiments are repeated with subsets of the AMAZON and WIKI data sets (i. e. considering only a part of all labels, as in Section 5.6—see there for further details of the setup), it transpires that at least with the AMAZON set compound resolution helps in a large majority of cases with up to 13 labels (both with and without stopwords). For the WIKI set, on the other hand, the results remain without a clear interpretation (see the detailed results in the Appendix, Tables D.26 and D.27).

6.2 Using Syntactic Information

In this section we look at techniques using syntactic information to generate new features. Typically these new features assume the form of a combination of existing features which are thought to belong together and convey information not present in the individual parts. Non-linguistic literature often refers to all such multi-term sequences as "phrases".

We begin by looking at bigrams, the most simple such "phrases" which do not require any syntactic knowledge (Section 6.2.1). Then we move on to multi-part names (Section 6.2.2) and noun phrases (Section 6.2.3).

All "phrases" discussed in this section make use of context information and thus they must be prepared at the *vectorisation* step (see Section 3.2.3). Each "phrase representation" was first evaluated on its own, and afterwards in combination with the BOL_{stop} model. For the new features we experimented with weighting factors of 1, 2, 3 and 10.

6.2.1 Bigrams

Using simple bigrams as features, either in place of or in addition to normal terms, is a relatively frequently adopted in practice (cf. Section 3.2.3.1). Here it is mainly used as a reference point for the methods discussed further below.

In detail, we followed these steps to create the bigram representation of each document:

1. Lemmatise all words (in the original order).
2. Remove all stopwords.
3. Build successive pairs of words (bigrams) not transgressing sentence boundaries.
4. Bring the members of each bigram into a standard order.[3]

Example (underlined words = words not on the stopword list):

> "Wie macht man sich das Leben leicht? Man schlafe bis Mittag, nehme ein gesundes Mahl zu sich und halte darauf bis zur Bettgehzeit ein erfrischendes Nickerchen."
>
> →
>
> (Leben, leicht) (Mittag, schlafen) (Mittag, gesund) (Mahl, gesund) (Mahl, halten) (Bettgehzeit, halten) (Bettgehzeit, erfrischend) (Nickerchen, erfrischend)

[3] Alphabetical order was used, with capital letters before lowercase letters, though any consistent sorting principle would have done. A cursory experiment had revealed that omitting this step and keeping the original order (i. e. distinguishing between [A,B] and [B,A]) was unlikely to improve the results.

It is evident that not all bigrams reach the same level of meaningfulness.

Table 6.3 lists the twenty most frequent such bigrams for each data set. It transpires, in particular when looking at the first two sets, that a few of the most frequent bigrams come from typical abstract expressions of either language or the particular domain in general, without contributing to the description of a particular feature or subject matter (e. g. Buch/behandeln or Auflage/neu). In fact, these are rather "stop phrases" than subject-specific collocations. We also notice a few irregularities caused by imperfections in the document preparation process such as "Systematik/ffc0c0" or "<br/clear" in the WIKI set.[4] In the AMAZON set a number of typical review sources show up which might have been skipped at the document cleansing step ("Neue Zürcher Zeitung", "Perlentaucher Medien GmbH", "Uni-Studentenrezension" and "Buch der 1000 Bücher", the last of which immediately produces *two* "1000/Buch" tokens, of course).

Looking at the outcome of the bigram experiments (Figure 6.4), we find that despite effectively increasing the number of shared features significantly compared to the BOL_{stop} representation, bigrams on their own lead only in one case (SDA) to semi-acceptable results. Similarly, also as *additional* features we find bigrams only useful for the SDA set; in the other cases they tend to make clustering results rather worse than better.

6.2.2 Multi-part Names

We would expect *proper names* (of people, companies, products, places, countries and regions) to be useful features for clustering. A first step towards stressing the importance of names had been our POS experiments wherein individual NAM-tokens were given special treatment (Sections 5.5 and 5.7.1). That approach had proven only partially successful. It is natural to expect results to improve if we turn our attention from simple name tokens to *multi-part* names, i. e. names consisting of several parts (such as a first and family name).

To this effect we collected name sequences from the input text by the following greedy algorithm:

1. Collect NAM, NAM_1 and NAM_2 tokens (including typical name particles such as "von") as well as abbreviations.[5]
2. Stop at the first occurrence of a token belonging to other classes or of a punctation mark (except commas[6]).
3. From the resulting name sequence eliminate:
   - foreign name particles ("of", "the", "and", "de", "le", "la", "with", "van", "et" and "San").
   - German name particles ("von", "der", "zu", "und", "auf", "zur").
   - Typical abbreviations ("Mrd.", "Mio.", "Tsd.", "Fr.", "Dr.", "Prof.", "Ing.", "Jur.", "Chr.", "ca.", "Nr.", "Jh." and all single letters).
4. Discard a sequence if it consists exclusively of abbreviations or lower-case letters.
5. Bring the tokens into standard (alphabetical) order and uniform capitalisation[7].

[4] On the other hand, despite looking unusual, the combination "Koordinate/Äquinoktium" is a perfectly legitimate feature which occurs in numerous astronomical Wikipedia entries where stellar bodies are identified by their coordinates in space.

[5] I. e. SUB tokens to which GERTWOL had added an "ABK" (abbr.) tag.

[6] An extra experiment had shown that adding commas to the name delimiters did not make much difference.

[7] The latter seems recommendable because names and abbreviations are particularly prone to individual and inconsistent capitalisation practices.

SPRINGER	AMAZON	SDA
Diagnostik/Therapie (79)	Buch/lesen (2269)	Jahr/vergangen (3940)
Buch/behandeln (79)	Geschichte/erzählen (1334)	Franke/Million (2863)
Buch/vermitteln (69)	Zürcher/neu (1283)	Nachrichtenagentur/sda (1852)
Grundlage/theoretisch (62)	Zeitung/Zürcher (1280)	Galle/St. (1735)
Entwicklung/neu (62)	Medium/©Perlentaucher (1269)	Mitteilung/heißen (1697)
Buch/beschreiben (61)	GmbH/Medium (1259)	Jahr/alt (1613)
Buch/enthalten (58)	Jahr/alt (1204)	Mensch/töten (1591)
Auflage/neu (55)	-Studentenrezension/Uni (1041)	Bush/George (1565)
Klinik/Praxis (54)	Frau/jung (1030)	Woche/vergangen (1496)
umfassend/Überblick (52)	20./Jahrhundert (919)	Mio/Prozent (1247)
aktuell/Überblick (48)	Buch/neu (882)	Kanton/Zürich (1201)
Buch/umfassend (48)	Buch/empfehlen (868)	Prozent/steigen (1113)
Erkenntnis/neu (47)	Frau/Mann (804)	Leben/Mensch (1095)
Schwerpunkt/bilden (46)	Fall/jed (796)	George/US-Präsident (1091)
Buch/liefern (44)	Buch/spannend (755)	Alte/Jahr (1091)
Darstellung/umfassend (43)	Buch/enthalten (749)	Euro/Mrd. (1085)
Buch/richten (43)	1000/Buch (702)	Ende/Jahr (1070)
diagnostisch/therapeutisch (40)	Hand/legen (676)	Galler/St. (1034)
Abbildung/Tabelle (40)	Zeitung/allgemein (665)	Bern/Kanton (1025)
Buch/Leser (39)	kennen/lernen (657)	Jahr/Prozent (1013)

WIKI	NZZ
Systematik/ffc0c0 (2292)	Jahr/vergangen (2754)
19./Jahrhundert (1438)	Franke/Million (2694)
20./Jahrhundert (1316)	Jahr/alt (1412)
Bundestag/deutsch (943)	Ende/Jahr (1402)
Politiker/deutsch (859)	Jahr/sechziger (1284)
Alte/Jahr (858)	Rolle/spielen (1228)
Rolle/spielen (805)	Staat/vereinigt (1192)
Jahr/alt (789)	Jahr/lang (1096)
Schriftsteller/deutsch (758)	Dollar/Million (1081)
Koordinate/Äquinoktium (689)	Clinton/Präsident (1072)
Bundesrepublik/Deutschland (638)	Woche/vergangen (1051)
	NZZ/Nr. (1008)
Mitglied/deutsch (598)	Union/europäisch (1005)
18./Jahrhundert (597)	Frau/Mann (993)
2000/Jahr (571)	19./Jahrhundert (991)
Jahr/lang (567)	Kanton/Zürich (972)
2000/Äquinoktium (534)	Stadt/Zürich (841)
1970er/Jahr (527)	Jahr/fünfziger (834)
Spiel/olympisch (525)	20./Jahrhundert (809)
1990er/Jahr (493)	Anfang/Jahr (787)
1980er/Jahr (478)	

Table 6.3: **The 20 most frequent bigrams** in each data set (in parentheses: document frequencies).

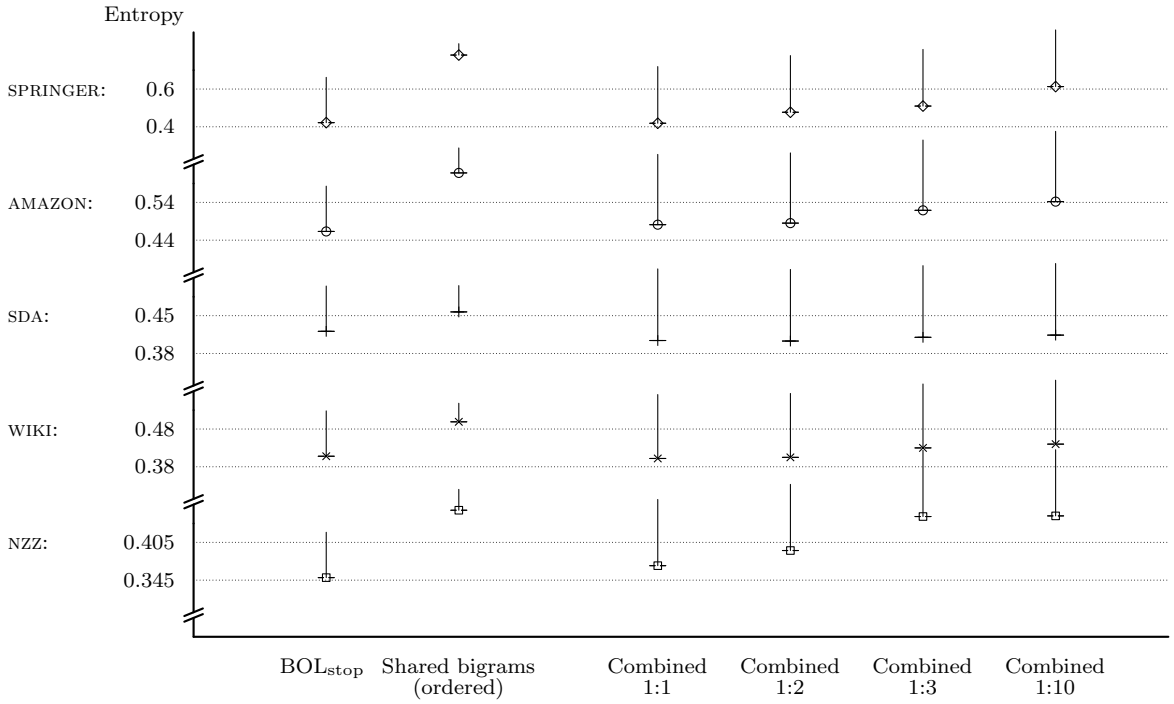


Figure 6.4: **Clustering with *bigram* features.** "Combined 1:10" means that the frequencies for the bigrams were multiplied by ten (before tf-idf weighting). The length of the vertical lines refer, as before, to the number of non-zero elements. (For the underlying numbers refer to Table D.28.)

SPRINGER	AMAZON	SDA
Österreich Schweiz (12)	New York (1009)	New York (1592)
Günther Winkler (9)	Mb Ram (424)	Bush George (1533)
Europa Usa (9)	Science Vol. (353)	Kanton Zürich (1054)
Eth Zürich (9)	Agatha Christie (345)	Bern Kanton (860)
New York (7)	Mann Thomas (298)	Hussein Saddam (658)
Commerce Electronic (5)	Angeles Los (212)	Annan Kofi (650)
Aachen Rwth (5)	Grimm Wilhelm (160)	Ariel Scharon (604)
Rudolf Schwarz (3)	Grimm Jacob (155)	Kanton Solothurn (578)
Japan Usa (3)	Hermann Hesse (148)	Arafat Jassir (561)
Janeiro Rio (3)	Harry Potter (145)	Gerhard Schröder (523)
Hans Scharoun (3)	Fontane Theodor (145)	Galle Kanton (452)
Donald Knuth (3)	Ost West (137)	Chirac Jacques (451)
Din En Iso (3)	Luther Martin (119)	Blair Tony (425)
Diener Diener (3)	Alfred Hitchcock (118)	Berlusconi Silvio (415)
Business Electronic (3)	Franz Kafka (115)	El Sadr (391)
Angeles Los (3)	Christus Jesus (112)	Colin Powell (375)
Aldo Rossi (3)	Adolf Hitler (110)	Hans-rudolf Merz (351)
Re Swiß (2)	Karl May (108)	Aargau Kanton (336)
New Paris York (2)	Astrid Lindgren (106)	Fdp Svp (276)
München Tu (2)	New Times York (100)	Kanton Luzern (273)

WIKI	NZZ
New York (1558)	New York (1511)
Angeles Los (460)	Kanton Zürich (901)
Christus Jesus (309)	Angeles Los (384)
König Portugal (216)	New Times York (254)
Friedrich Wilhelm (207)	Bern Kanton (253)
König Schweden (200)	Boutros Ghali (237)
König Spanien (197)	Eth Zürich (236)
Immanuel Kant (195)	Ost West (230)
Luther Martin (184)	Janeiro Rio (209)
China Kaiser (175)	Hussein Saddam (199)
Jersey New (171)	Aviv Tel (171)
Japan Kaiser (167)	Aires Buenos (162)
König Polen (166)	Berlusconi Silvio (142)
Adolf Hitler (160)	Mann Thomas (141)
Dänemark König (157)	Aargau Kanton (129)
Burundi König (157)	Paulo São (128)
Goethe Johann Wolfgang (152)	Johannes Paul (125)
Johannes Paul (146)	John Major (121)
Shakespeare William (144)	Felipe González (118)
Karl Marx (139)	Amnesty International (114)

Table 6.4: **The 20 most frequent multi-part names** in each data set.

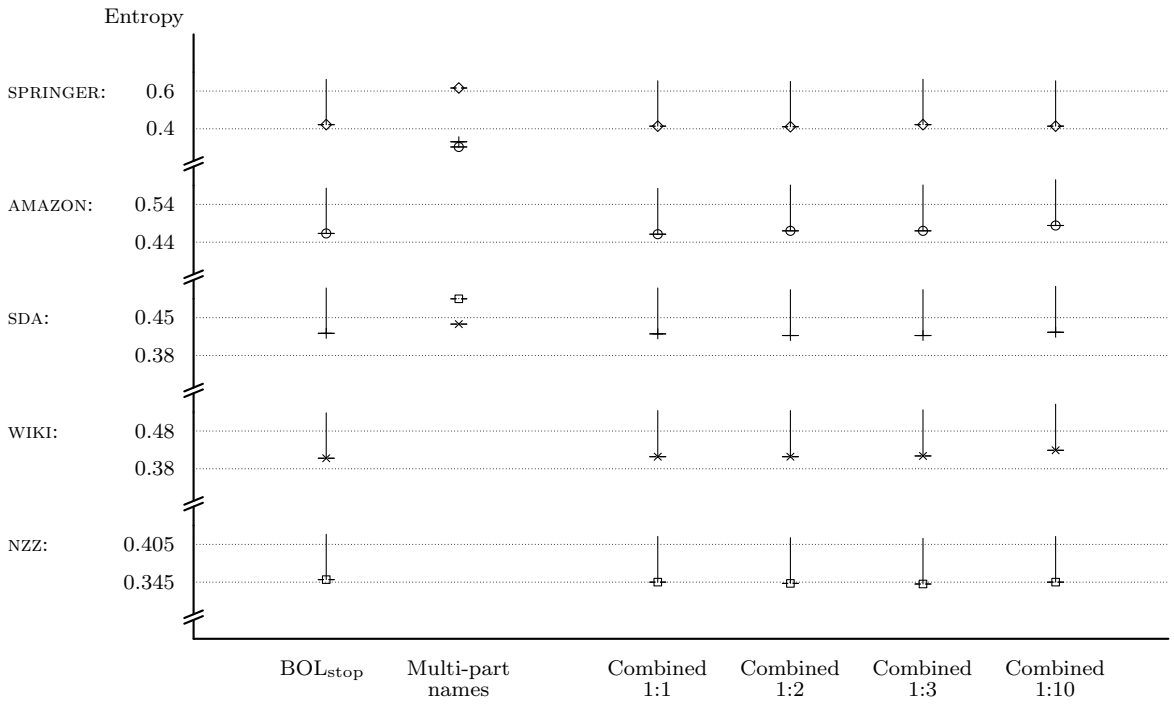


Figure 6.5: **Clustering with *multi-part name* features.** Since multi-part names were rare (0.08–1.71% of the number of ordinary features), the representation relying solely on them (second column) was unsatisfactory (with 3,721–39,661 documents not being clustered at all). (For the underlying numbers refer to Table D.29.)

	SPRINGER	AMAZON	SDA	WIKI	NZZ
Fully parsed	94%	91%	96%	68%	92%
Majority of text parsed	5%	7%	4%	24%	5.8%
Majority of text not parsed	0.5%	1.5%	0%	3%	1.5%
Not parsed at all	0.5%	0.5%	0%	5%	0.7%

Table 6.5: **Degree of parsing success.** Extent to which the documents in each data set were parsed using Gojol's dependency parser.

Table 6.4 shows the twenty most frequent multi-part names found for each data set. We encounter mostly well-known and sensible named-entity concepts, even though the inclusion of abbreviations has produced features such as "MB RAM" and "Science Vol.". Even though they do not fulfil our expectation of a proper name, there is still a strong link between the two parts of such pairs, suggesting that the combination is hardly harmful. Some further anomalies are caused by GERTWOL's tagging in some contexts "Kanton", "König", "Kaiser" as proper names, which at least in a narrow sense they are not. Again, however, the resulting features (e. g. "China/Kaiser", "Kanton/Solothurn") are often very sensible features conveying extra information not expressed in the parts.

The outcome of the clustering experiment with multi-part names is shown in Figure 6.5. Evidently, multi-part names are far too rare to adequately represent the documents on their own, but as additional features they can perhaps prove useful, though the effect remains small. It emerges that the best results are achieved by giving extra, perhaps double, weight to the multi-part names, with the usual pattern repeating itself: SPRINGER, SDA and NZZ tend to profit from the extra features whereas the opposite is true for AMAZON and WIKI.

Turning again to the lists of the twenty most frequent names, readers acquainted with the respective corpora/subjects will easily recognise most of the names as valid features for all five data sets. However, it might be argued that although many of the names may be useful as cluster descriptors (cf. Section 3.6.2), they add only little information not already contained in the individual parts. For instance, the feature "Buenos Aires" will probably have almost the same distribution as "Buenos" and "Aires" alone, rendering the composite feature quite superfluous. Still, this does not explain why the results are getting almost immediately worse when these features are added to AMAZON and WIKI .

We may conclude that at least with our simple, domain-independent multi-part name recogniser, the positive impact of multi-part name features is minimal, with the most suitable corpus being the NZZ set. Perhaps a more sophisticated proper name recognition system may improve results, although a significant difference would appear rather unlikely to us.

6.2.3 Noun Phrases

We now turn to "real" phrases, i. e. phrases in a linguistic sense. For this purpose Vlad V. Gojol's dependency parser was used (for a brief description see Clematide, 2002). Since the program proved not very stable with regard to irregularly formatted input, not all texts were parsed in full. Table 6.5 shows the extent to which each data set had been parsed.

From the available parses we then extracted *noun phrases (NPs)* and standardised them as follows:

1. Resolve nested phrases. E. g. "Der lange Schatten des kleinen Mannes" is turned into three phrases: "Der lange Schatten", "des kleinen Mannes" and the original "Der lange Schatten des kleinen Mannes".
2. Keep nouns, names, adjectives and verbs (i. e. those word carrying GOJOL's N, V and ADJA tags). Discard all other tokens.
3. Discard all stopwords.
4. Stem all remaining terms and bring them into alphabetical order.[8]
5. Remove any phrases consisting of just a single word and eliminate duplicates arising from the same initial phrase.

Table 6.6 lists the most frequent noun phrases in each data set. We easily recognise quite a few typical and useful phrases, but also a preponderance of temporal specifiers of limited worth for clustering ("laufend/jahr", "vergang/woch" etc.) which occur mostly in the news sets SDA and NZZ, but also in WIKI. An extended, domain-specific stopword or "stop phrase" approach might be useful in order to avoid these phrases. However, in some other situations they might actually be useful, so it is dangerous to drop them regardless of the context (corpus).

The clustering results as shown in Figure 6.6 indicate that noun phrases can be useful additions indeed (SDA and in particular NZZ). For the other sets the effect tends to be negative, however. Once more we find that adding useful extra information to the AMAZON and WIKI $\text{BOL}/\text{BOL}_{\text{stop}}$ models is extremely hard.

6.2.4 Conclusions

Our three experiments in this section (bigrams, multi-part names and noun phrases) made it clear that enhancing the BOL_{stop} model for clustering is not at all trivial. Using syntactic information to create new features mostly failed to bring special dividends, the only exception being the NZZ set, where both multi-part names and noun phrases were found useful to a degree not encountered in the previous chapter. A natural hypothesis is that richer and longer texts (as in the NZZ data set) lend themselves better to syntactic methods than shorter and less well structured texts, but the present basis of just five data sets does not allow us to draw any definite conclusions.

The usefulness of "phrases" as *cluster descriptors* has not been examined in this study. It would not be surprising, however, if in particular multi-part names would turn out to be good cluster descriptors, even in situations where they fail to improve the actual clustering quality.

[8] For practical reasons this solution was preferred to the lemmata because the parser output does not correspond to the input text in strict linear order.

SPRINGER	AMAZON	SDA
grundlag/theoret (53)	medi/perlentauch (1265)	jahr/vergang (3829)
umfass/überblick (50)	neu/zeitung/zürcher (1222)	vergang/woch (1537)
entwickl/neu (46)	studentenrezension/uni (985)	jahr/laufend (1156)
erkenntniss/neu (38)	frau/jung (751)	euro/mrd (1019)
praxis/täglich (35)	buch/bücher (735)	frank/mrd (941)
klinik/praxis (35)	süddeutsch/zeitung (568)	mio/prozent (935)
darstell/umfass (34)	allgemein/zeitung (521)	dollar/mrd (857)
diagnost/therapi (32)	buch/neu (478)	europä/union (823)
aktuell/entwickl (32)	buch/gut (476)	fr/mrd (746)
buch/ziel (31)	freier/schriftstell (457)	monat/vergang (665)
arbeit/täglich (30)	ganz/welt (438)	nation/vereint (664)
maßstäb/neu (27)	jung/mädchen (430)	kanton/zürich (657)
abbild/tabell (27)	buch/end (414)	anfrag/nachrichtenagentur (505)
europä/union (26)	deutsch/sprach (409)	tag/vergang (503)
deutsch/gesellschaft (26)	jung/mann (379)	verkehr/öffentlich (491)
anwend/praktisch (26)	mb/ram (372)	arme/israel (471)
aufschlussreich/text (25)	erwachs/kind (360)	angeleg/auswärtig (451)
praxis/wissenschaft (24)	leb/neu (352)	schwer/verletz (435)
geschicht/kult-objekt (24)	geschicht/spannend (342)	europä/zentralbank (424)
form/funktion/markenkommunikation/maßstäb/neu (24)	neu/roman (329)	mio/verlust (420)

WIKI	NZZ
bundestag/deutsch (840)	jahr/vergang (2666)
deutsch/polit (599)	staat/vereinigt (1160)
koordinat/rektaszension/äquinoktium (593)	jahr/sechzig (1127)
	vergang/woch (1031)
deutsch/schriftstell (529)	europä/union (973)
hellig/scheinbar (514)	jahr/laufend (893)
gleich/jahr (507)	nr/nzz (853)
olymp/spiel (505)	fünfzig/jahr (769)
1970er/jahr (429)	jahr/neunzig (682)
bundestag/deutsch/mitglied (415)	nation/vereint (652)
1990er/jahr (410)	jahr/letzt (597)
kathol/kirch (405)	stadt/zürich (584)
1980er/jahr (397)	kanton/zürich (555)
end/jahrhundert (376)	hand/öffentlich (508)
1960er/jahr (362)	bosnisch/serb (507)
dat/technisch (358)	dreissig/jahr (491)
staat/vereinigt (353)	kalt/krieg (489)
gregorian/kalend (352)	ganz/land (480)
deutsch/schauspiel (349)	halb/jahr (449)
gregorian/kalend/tag (333)	jahr/zwanzig (433)
deutsch/sprach (329)	

Table 6.6: **The 20 most frequent noun phrases** in each data set.

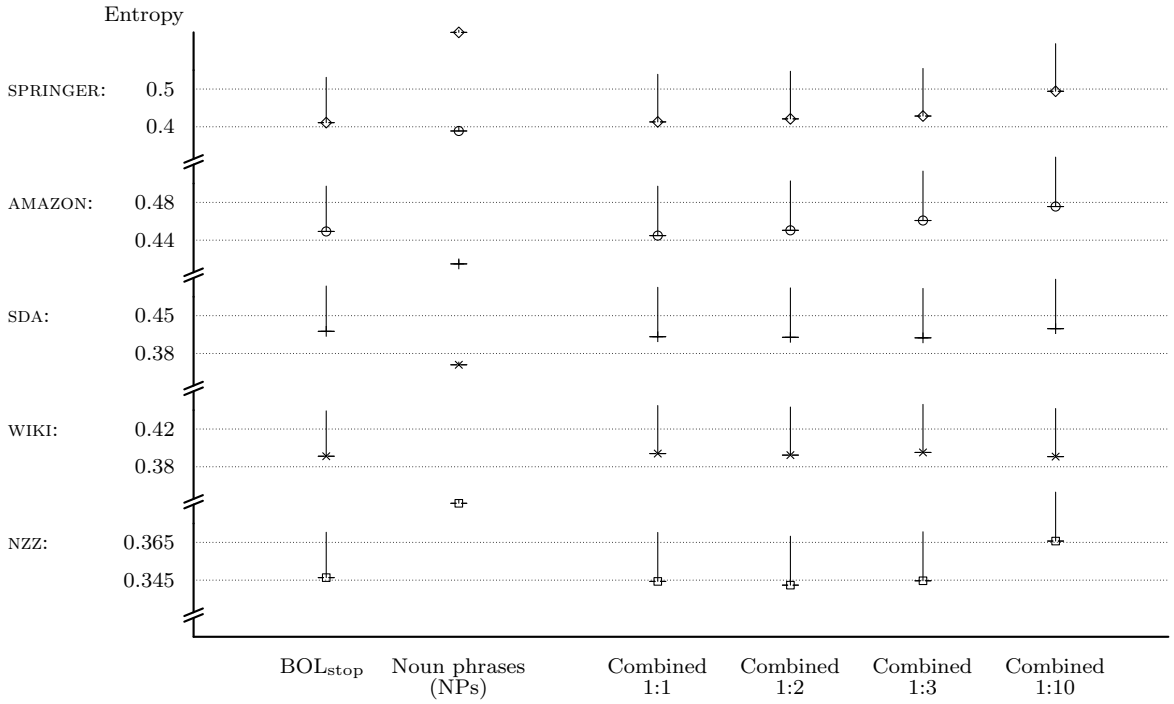


Figure 6.6: **Clustering with *noun phrases*.** The number of common noun phrases being relatively rare, between 424 and 6,312 documents could not be clustered in the second column of the figure. These documents had thus no NPs in common with other documents. (For the underlying numbers refer to Table D.30.)

6.3 Using Semantic Information

In a perfect world we would choose to represent each document not by its words and phrases, but by its content, i. e. in terms of *semantic* concepts. However, for open domains and documents without a uniform structure such a representation is unfeasible, at least for the time being. What we can do is try to make use of semantic knowledge that is typically stored in ontologies. This is not sufficient for a full-fledged semantic representation, but at least it allows us to recognise certain closely related concepts (words) and represent them as single features.

Section 6.3.1 introduces the ontology used for our experiments, Section 6.3.2 deals with the problem of finding the right concept for each term and Section 6.3.3 describes the actual usage and the experimental results. Two further sections are devoted to refinements and a brief summary.

6.3.1 GermaNet

GermaNet (Hamp and Feldweg, 1997) is a lexical resource for German developed at the University of Tübingen, Germany and akin to the English WordNet (Miller *et al.*, 1990). Sets of lemmatised synonyms and near-synonyms (so-called "synsets") lie at the bottom of GermaNet. However, in addition to these synonym relations, a number of further semantic relations are modelled between these synsets:

Antonyms. The opposite of synonyms, e. g. "warm" and "kalt" (hot and cold).

Hyper- and hyponyms. Hierarchical relations between similar concepts of different degrees of specificity. E. g. "Hund" (dog) is a hyponym of "Haustier" (domestic animal) and the latter a hypernym of the former.

Mero- and holonyms. Relations between concepts of which one forms part of another. E. g. "Arm" is a holonym of "Hand" and the latter a meronym of the former.

Pertainyms. Relations between denominal adjectives and their nominal bases, deverbal nominalisations and their verbal base, deadjectival nominalisations and their adjectival base. E. g. "fehlerlos" (flawless) and "Fehler" (flaw).

Entailment. Relations between two verbs of which one entails the other. In our version of GermaNet only a few examples were encoded. E. g. "erwarten" (to expect, in the sense of expecting a baby) entails "zeugen" (to beget).

Cause. The cause relation holds between verbs (114 cases) and between verbs and resulting adjectival states (95 cases). E. g. "einladen" (to invite) and "besuchen" (to visit) or "öffnen" (to open) and "offen" (open).[9]

Table 6.7 reports on the amount of data and relations available in GermaNet, version 4.0.

In order to achieve balanced and logical hypernym hierarchies a number of artificial concepts was introduced such as "angestellter Mensch" (employed person) for which there exists no proper and exact term in German. No global hyponym hierarchy exists; instead, several independent sub-hierarchies exist (which can also contain cross-references).

Besides, a number of short phrases has also been included in GermaNet. We have not tried to make use of these in our experiments.

[9]However, it appears to us that many of the relations between verbs are of limited usefulness. E. g. "zersumpfen" → "sumpfen", "zerdreschen" → "dreschen", "bremsen, zügeln" → "verlangsamen" or, indeed, "leeren" → "leeren".

	nouns	adjectives	verbs	total
Single terms	34,501	6,886	7,742	49,129
Short phrases	2,110	80	203	2,393
Artificial concepts	212	116	94	422
Relationships among terms and phrases:				
Synsets	27,241	5,106	8,733	41,080
Antonym pairs	605	500	235	1,340
Hyper-/hyponym relations	30,075	4,997	9,202	44,274
Mero-/holonym relations	3,914	—	—	3,914
Pertainyms	8	1,515	132	1,655
Entailment relations	—	—	7	7
Cause relations	—	—	209	209

Table 6.7: GermaNet **4.0 statistics.** The two adverbs forming a separate class have been omitted. Of the nouns, 1,712 represent proper names. "Short phrases" describes entries consisting of more than a single term (e. g. "wilde Tulpe").

It should be noted that a single term can be part of several synsets—depending on its different meanings. The climax of *polysemy* is reached by the 26 senses encoded for "halten" (to hold), followed by "kommen" (to come) with 18 senses. However, the large majority belongs to only one (90%) or two synsets (7%).

6.3.2 Word Sense Disambiguation

For the ten percent of words that can have multiple senses, the classical *word sense disambiguation* problem arises. Despite substantial efforts (see, for instance, Schütze, 1998; Merlo *et al.*, 2003; Stevenson, 2003) it still poses many difficulties. The two main approaches to tackle the problem are based on co-occurrence statistics and knowledge sources (Leacock *et al.*, 1998).

For our purposes (distinguishing between multiple synset candidates) we must rely on the information available from GermaNet. For each sense of a given polysemous word in GermaNet we extract a "support set". The support set consists of all terms connected with that sense, i. e. all members of a given synset as well as all its antonyms, holonyms, hypernyms, meronyms, etc. If a polysemous word occurred in a document, we then looked at each possible sense and calculated its support by counting the occurrences of the support set members in that document. In case of a tie the sense with the lower number as encoded in GermaNet is used. The sizes of the support set may vary substantially for different senses, thus giving certain senses more opportunities for "scoring". However, based on the assumption that better connected synsets were also more frequent and thus more likely to correspond to the senses actually sought, this inequality was not considered a serious flaw.

While this method seems to work to a certain degree, it should also be mentioned that often the support sets remain empty for all candidates. In the absence of GermaNet-specific, large annotated corpora we had to be satisfied with this rather basic disambiguation method.

6.3.3 Semantic Mapping

For our document clustering experiments, we tried to exploit the ontological information provided by GermaNet in different ways. At the core of all experiments lie the synsets, with different

mapping variants being characterised by different uses of the additional relationships outlined in Section 6.3.1 above and by different disambiguation strategies:

as: Map each lemma to all its synsets (where available).

fs: Map each lemma to its first synset (as defined in GermaNet).[10]

fs+cep: Map each lemma to its first synset, but replace all synsets in a pertainym, entailment or cause relationship by the corresponding more general synset.

fs+o: Map each lemma to its first synset and merge opposites, i. e. synsets in an antonym relation with one another.

fs+hy[n]: Map each lemma to its first synset. Then replace all "leaves" in the hyper-/hyponym hierarchies by the next higher synset(s) (i. e. replace all synsets with no hyponyms by their hypernyms). Execute this step n times. The intention is to use only those relations that occur at the bottom of the hierarchies where terms have a specific nature. Clifton *et al.* (2004) aim for a similar effect by traversing the hierarchies top-down and retaining just those relations that occur at level five or higher.

fs+ho: Map each lemma to its first synset. For all synsets with a holonym, *add* the synset frequency value to that holonym.

ds: Disambiguate word senses with support sets (cf. Section 6.3.2) and choose the most likely synset for each lemma.

The results (Figure 6.7) do not suggest real improvements in clustering quality with any of the methods tested. In particular, using all senses of a given word (**as**) makes the results significantly worse. Using hypernym relations (**hy**) is occasionally successful, but it is not clear under which circumstances and there is a danger of the results turning worse as well. See, for instance, the widely varying results of the springer set with different integration levels (**hy**[1], **hy**[2], **hy**[3]).[11]

In order to better understand the effects of sense mapping, Tables 6.8 and 6.9 list the twenty most frequent synsets for each data set under model **fs**. At least from an intuitive point of view, the large majority of these most frequent "unifications" do not seem helpful for our purposes (i. e. they do not help distinguishing actual document topics). On the contrary, they tend to increase similarity between documents based on the use of more or less functional words (for instance, "Thema"/"Schwerpunkt"). Since the distribution of such expressions has often little in common with that of the main topics in a collection, these synsets must have rather a detrimental effect on the clustering results. We also note a number of synsets which a proper word disambiguation would avoid (e. g. "Vater"/"Herr", "Fall"/"Deklination") and synsets which seem of doubtful values such as "Antrag"/"Angebot". On the other hand, pairs such as "Team"/"Mannschaft", "Uni"/"Universität" and "festnehmen"/"verhaften" are very likely to increase clustering performance.

[10] Unfortunately, unlike in WordNet the different senses in GermaNet are not ordered by frequency. This leads to first senses such as "god" for "father", "Jesus" for "son", and so on. Sticking to the first synset available is thus less satisfactory than in the study of Clifton *et al.* (2004) who exploit the fact that in the English WordNet the senses are ordered by frequency and that in about 80% of all cases the most frequent synset is indeed the one actually sought.

[11] Cf. also the **hy/N** experiments in Table D.32 in the Appendix which confirm the volatility of the hypernym experiments.

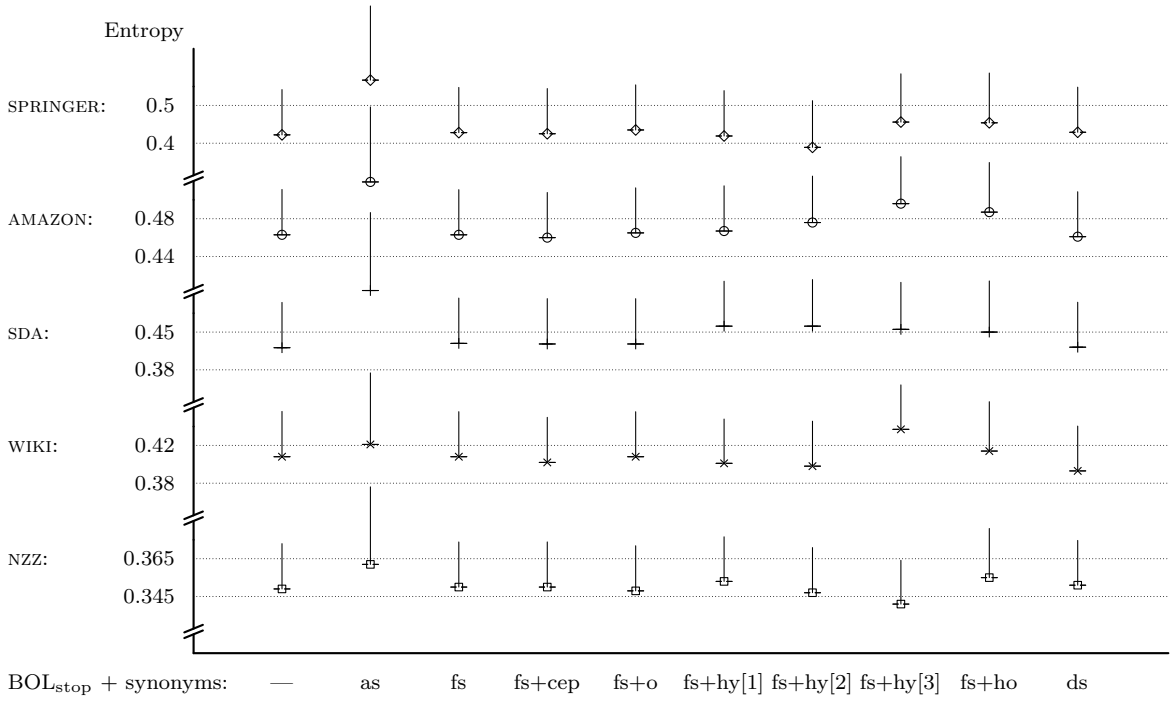


Figure 6.7: **Clustering with synsets.** (For the underlying numbers refer to Table D.31.)

SPRINGER	AMAZON	SDA
[thema(442) / schwerpunkt(258) / motto(4)]	[autor(10788) / autorin(4120) / verfasser(667) / verfasserin(104)]	[mensch(8505) / person(5971) / persoenlichkeit(370) / individuum(55)]
[bereich(387) / gebiet(204)]	[mensch(9235) / person(2480) / persoenlichkeit(1140) / individuum(203)]	[bereich(3535) / gebiet(1956)]
[ueberblick(359) / uebersicht(124)]	[freund(5891) / freundin(2104) / liebhaber(567) / liebhaberin(24)]	[anfang(4362) / beginn(1888) / anbeginn(3)]
[wissen(202) / kenntnis(113)]	[geschichte(15837) / vergangenheit(2055) / historie(199)]	[antrag(1777) / angebot(1679) / offerte(143) / avance(28)]
[oekonomisch(125) / wirtschaftlich(111)]	[ende(6377) / schluss(1826) / endpunkt(29) / schlusspunkt(20) / aus(3)]	[erhoehen(3051) / steigern(1576)]
[praktisch(405) / praxisorientiert(104) / praxisbezogen(93)]	[vorstellen(2493) / praesentieren(1612)]	[brand(1589) / feuer(1376)]
[vorstellen(180) / praesentieren(84)]	[bereich(2414) / gebiet(1553)]	[platz(1799) / raum(1361)]
[diskutieren(170) / eroertern(76) / besprechen(27)]	[anfang(3124) / beginn(1362) / anbeginn(36)]	[ende(9285) / schluss(1243) / schlusspunkt(36) / endpunkt(3)]
[notwendig(176) / erforderlich(68) / unerlaesslich(22)]	[raum(1484) / platz(1318)]	[ermoeglichen(1298) / erlauben(1201)]
[methode(428) / vorgehen(65) / vorgehensweise(64)]	[vater(3748) / herr(1303)]	[sprecher(4291) / sprecherin(1189)]
[nutzen(67) / nutzung(64) / benutzung(7)]	[thema(6943) / schwerpunkt(1069) / motto(368)]	[ausloesen(1253) / verursachen(1038) / hervorrufen(88)]
[durchfuehrung(81) / realisierung(58) / verwirklichung(6)]	[wissen(2160) / kenntnis(1055)]	[stelle(2711) / position(1015) / anstellung(113) / posten(23)]
[ermoeglichen(218) / erlauben(54)]	[ausgabe(1859) / edition(1054)]	[protest(1015) / beschwerde(1009) / beanstandung(56) / reklamation(28) / anfechtung(13)]
[experte(88) / fachmann(53) / expertin(2)]	[uni(1117) / universitaet(1049)]	[praesentieren(1565) / vorstellen(990)]
[mensch(122) / person(51) / persoenlichkeit(16) / individuum(8)]	[wirklichkeit(1196) / realitaet(993)]	[besuch(1320) / besucher(957) / besucherin(295)]
[diagnose(72) / befund(46)]	[faszinierend(2022) / fesselnd(989) / begeisternd(43)]	[sitzung(986) / konferenz(932)]
[autor(642) / autorin(45) / verfasser(30) / verfasserin(2)]	[darstellung(3045) / schilderung(948)]	[festnehmen(2236) / verhaften(909) / greifen(850) / inhaftieren(255) / arretieren(3)]
[fortschritt(64) / weiterentwicklung(45)]	[rat(923) / tip(845) / ratschlag(38)]	[anstieg(1087) / zunahme(849) / ansteigen(15)]
[veraenderung(89) / aenderung(43)]	[tat(934) / verbrechen(745) / straftat(22) / delikt(8)]	[einwohner(1004) / bewohner(818) / einwohnerin(113) / bewohnerin(109)]
[fachgebiet(76) / fachrichtung(41) / fachbereich(9) / sachgebiet(2)]	[eindrucksvoll(868) / beeindruckend(742)]	[einschaetzung(991) / schaetzung(748)]

Table 6.8: **The 20 most frequently used synsets** (under the **fs** mapping) for each data set (ordered by the occurrences of the second most frequent member in each synset).

WIKI	NZZ
[mensch(5207) / person(2824) / persoenlichkeit(593) / individuum(479)]	[mensch(6283) / person(5693) / persoenlichkeit(1237) / individuum(537)]
[bereich(3461) / gebiet(2543)]	[anfang(6259) / beginn(4130) / anbeginn(66)]
[anfang(3007) / beginn(1978) / anbeginn(41)]	[bereich(5491) / gebiet(4106)]
[platz(2059) / raum(1942)]	[platz(4664) / raum(4023)]
[zahl(2405) / anzahl(1899)]	[stelle(4143) / position(3230) / anstellung(179) / posten(55)]
[eigenschaft(2376) / merkmal(1509) / attribut(186)]	[ende(11872) / schluss(2861) / schlusspunkt(106) / endpunkt(47)]
[ermoeglichen(2206) / erlauben(1465)]	[geschichte(5626) / vergangenheit(2780) / historie(171)]
[benutzen(2372) / brauchen(1314) / gebrauchen(115) / benuetzen(66)]	[erlauben(2618) / ermoeglichen(2533)]
[vater(2481) / herr(1220)]	[praesentieren(3071) / vorstellen(2241)]
[stelle(2074) / position(1206) / anstellung(118) / posten(8)]	[besucher(2203) / besuch(2121) / besucherin(92)]
[herstellung(1171) / produktion(1027)]	[gegensatz(3221) / gegenteil(1721)]
[schauspieler(1286) / schauspielerin(920)]	[kontakt(2140) / umgang(1623)]
[zeichen(1150) / symbol(881)]	[aenderung(1756) / veraenderung(1607)]
[ueberwiegend(1121) / vorwiegend(856)]	[aktivitaet(1650) / taetigkeit(1592)]
[veraenderung(1028) / aenderung(836)]	[team(2253) / mannschaft(1586)]
[notwendig(1692) / erforderlich(821) / unerlaesslich(60)]	[traditionell(3098) / konservativ(1568) / restaurativ(9)]
[mannschaft(817) / team(804)]	[verstaerken(1958) / heben(1539) / mehren(303) / vermehren(155)]
[fall(4064) / deklination(801) / kasus(38)]	[einwohner(1625) / bewohner(1523) / bewohnerin(63) / einwohnerin(24)]
[bekannt(6777) / populaer(789)]	[realitaet(1592) / wirklichkeit(1509)]
[wirkung(1410) / effekt(786)]	[vater(2096) / herr(1491)]

Table 6.9: **The 20 most frequently used synsets** (under the **fs** mapping) for each data set (ordered by the occurrences of the second most frequent member in each synset).

6.3.4 Restrictions on Semantic Mapping

From the above observations it can be concluded that semantic unification is helpful in some cases and harmful in others. The present section presents a few attempts to narrow the selection down to "useful" synsets.

Apart from the semantic relationships in which they take part (such as holonymy), synsets can be characterised by their POS classes, by the number of their members, by their frequencies, by the frequencies of the individual terms and by the polysemy of the terms belonging to the synset. Apart from the number of members in a synset, all of these properties could provide clues for a successful sub-selection of all synsets. Four restriction techniques were thus tested:

[N|A|V]: restrict synset mapping to certain POS categories (assumption: not all word classes are equally good candidates for synset unification),

a[n]**:** restrict synset mapping to terms taking part in at most n synsets (since with too many options, word sense disambiguation is less likely to work and over-generalisation is likely to creep it),

df[i‰]**:** restrict synset mapping to terms whose document frequency does not exceed a certain limit (in analogy to compound splitting, we are more interested in bringing comparatively rare features together),

t[i‰]**:** restrict synset mapping to synsets whose document frequency will not succeed a certain limit.

In addition we tried to transfer the concept of stopwords to synsets, identifying "stop synsets" from the other four data sets according to the E'' measure (cf. Section 5.4.2):

su[n]**:** exclude stop synsets, gained by *combining* the top n stop candidates from the other four data sets (according to the E'' measure),

si[n]**:** exclude stop synsets, gained by *intersecting* the top n stop candidates from the other four data sets (according to the E'' measure).

Finally, for the hypernym experiments (**hy**[n]) we tried a similar restriction as Clifton *et al.* (2004):

m[n]**:** restrict synset mapping to integrated synsets (i. e. synsets and their hypernyms) with not more than n members (synonyms and hyponyms), an attempt to avoid over-generalisation.

The results of a few selected such experiments can be found in Figure 6.8, while the complete results are given in the Appendix (Table D.32). A certain tendency towards improvement can be observed, but there is no systematic pattern. Setting a limit on the document frequency of the partaking terms (**df**) appears to be the most promising strategy, relatively speaking. For the SPRINGER and WIKI sets the "stop synset" approach (**su**/**si**) also appears capable of improving results, and for SPRINGER and SDA the upper limit for synset frequency (**t**).

However, all in all, none of the restrictions promises a reliable separation of the good from the bad.

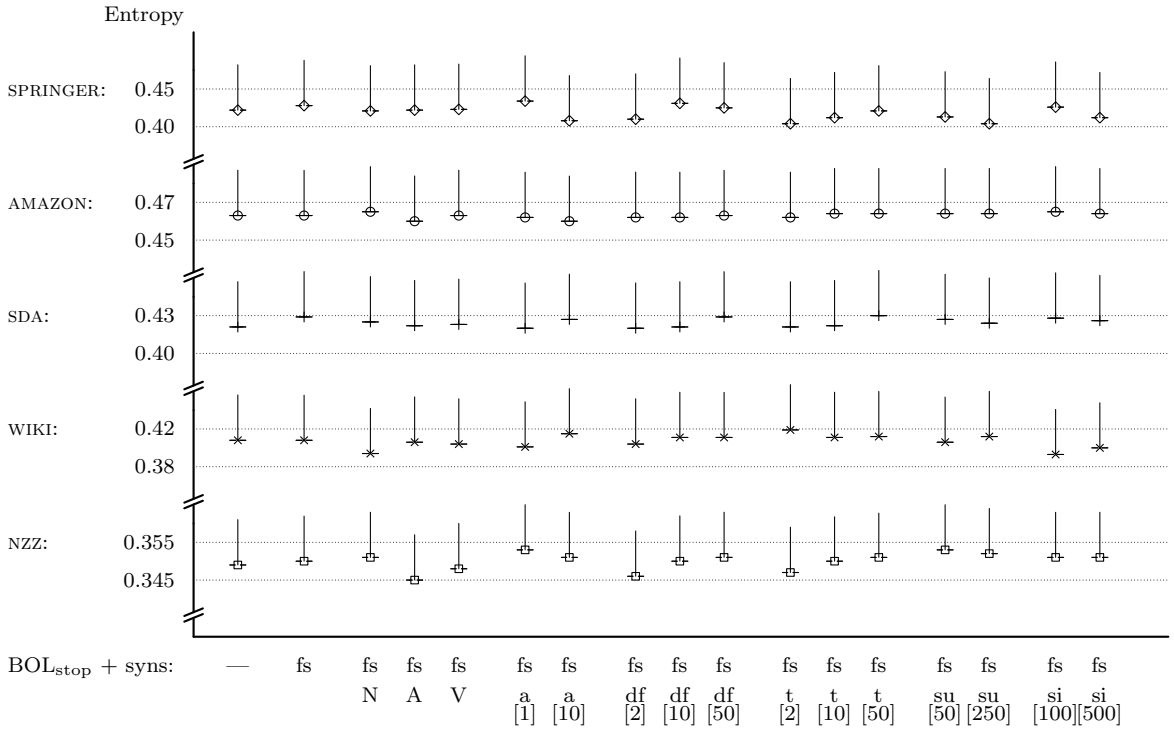


Figure 6.8: **Clustering with refined synset selection** (selected results, for the complete list see Table D.32 in the Appendix).

6.3.5 Conclusions

Using semantic information (in form of the GermaNet ontology) has after all failed to fulfil our expectations. Only in a few exceptional cases did the clustering results improve by the use of synsets, whereas in several other cases the results declined. Attempts to filter out those synsets that are blurring the distinctions between different topics (rather than accentuating similarities between related documents) were equally fruitless.

It appears that general ontologies contain too much information on unspecific concepts which outweigh the positive effects gained by identifying semantically close topical terms. Further research might try to overcome this problem, but we remain sceptical. A few complimentary experiments with "selected" synsets (i. e. synsets which were chosen either manually or according to one of the measures used for stopword extraction) confirmed the difficulties; even under such "artificial" circumstances the results did not differ markedly from those reported here.

6.4 Summary of Enhancement Experiments and NLP

We have examined techniques to enhance and improve document representations on three different levels: morphological, syntactic and semantic. Success varied by a marked degree.

Working with semantic representations has proved to be the most difficult approach and at least with our present tools it has proved a failure. Further research in that direction will have to concentrate on the difficult distinction between useful and useless semantic concepts (in terms of topic characterisation).

The verdict for multi-word features based on syntactic analysis is not too different. However, positive tendencies could be observed with the nzz set, prompting us to hypothesize that the usefulness of such methods may depend on the length and richness of the individual texts.

Finally, morphological analysis for compound splitting was found to be by far the most effective document enhancement method, with often large positive effects on clustering results. Of course, the usefulness of morphological analysis is connected to the tendency of a language to build compounds and inflect word forms. To what extent the success can be transferred to languages other than German must remain open.

Throughout this chapter we have, again, observed that the different data sets show different behaviour and that results of coarse clustering tasks (with few clusters) are more volatile and more easily improved than the fine-grained clustering tasks of the amazon and wiki data sets with their 21 resp. 22 labels. Concrete reasons for this rather counter-intuitive fact have not been found.

Chapter 7

Combining Document Representation Techniques

In the present chapter
I shall consider
the part which crossing plays in
two opposed directions:
firstly, in obliterating characters,
and consequently in preventing
the formation of new races;
and secondly, in the modification of old races,
or in the formation of
new and intermediate races,
by a combination of characters.
I shall also show that certain characters
are incapable of fusion.

Charles Darwin (*The Variation of Animals and Plants under Domestication*, 1868)

Although it is long, long way from evolutionary theory to document clustering, there are some parallels. As we have already seen in Chapter 2 cluster analysis has its roots in biological classification tasks, and both evolution and clustering can be viewed as optimisation processes working with masses of individuals with large numbers of features. And just as in evolutionary theory the question arises whether two individuals can match and produce an offspring that cultivates the best features of both, the same can be asked of document representation techniques: which can be combined to improve clustering results and which fail to work together?

The present chapter tries to answer this question by the examination of a view selected feature representation techniques, aiming to integrate the major findings of the previous two chapters and concentrating specifically on the natural language processing techniques. After the difficulties already encountered with various of the document processing techniques and the sometimes inconsistent behaviour for one and the same method (remember, for instance, the inconclusive evidence for stopword removal, which proved beneficial for some data sets and harmful for others), the expectations must naturally not be set too high.

7.1 Combining Matrix Reduction Techniques

From the reduction techniques (Chapter 5) we selected four (six variants in all) which had shown promising results if used on their own:

- **Stopword removal** with our standard stoplist;
- **POS selection**, with two variants: once with SUB and ADJ (= **pos1**) and once with SUB, ADJ and NAM_{all} (= **pos2**);
- **POS weighting**, with the same two variants (**wgt1**=SUB/ADJ and **wgt2**=SUB/ADJ/NAM_{all}) and weighting factor 1.5;
- **Pruning**, where individual "best" parameters were used. Concretely: **global** pruning for SPRINGER (0.05–10%), SDA (0.005–1%) and NZZ (0.01–10%) and **local** pruning for AMAZON ($\alpha = 150$) and WIKI ($\alpha = 100$).[1]

The results with different combinations of these techniques are given in Figure 7.1.

The emerging picture is similarly diverse as that of the individual techniques:

Stopword removal remained good for SDA and harmful for AMAZON and WIKI , with relatively little impact on SPRINGER and NZZ.

Pruning led to a similar picture, though the negative impact on AMAZON and WIKI remained very small, while the effects on NZZ set went in both directions.

POS selection proved useful for SPRINGER, but showed negative tendencies for AMAZON, WIKI and NZZ, whereas for SDA it did not produce the same good results in combination than when used as the only reduction technique.

POS weighting led to worse results for AMAZON and WIKI, whereas the positive effects on SPRINGER and SDA remained small. Only the NZZ set consistently seemed to profit from POS weighting.

If we compare these inconclusive verdicts with the results of the techniques used individually, we can at least note that the combined effects seem to stand more or less in relation with the individual effects. For instance, for the AMAZON set we had found that most techniques on their own worsened clustering quality. Their combination seems to have an even stronger negative effect. For WIKI pruning was the best method on its own, and also in the combinations it is the one that performs best, relatively speaking. For SPRINGER POS selection was the best individual technique and so it was in the combinations, whereas for NZZ the same can be said of POS weighting. Rather unexpected is only the relative disappointment of POS selection in combination with pruning or stopword removal on the SDA set.

We thus conclude that the reduction techniques behave more or less rationally. Techniques that have been found good for a given set individually, are also likely to produce satisfactory results in combination. On the other hand, techniques that were unsuccessful on their own seldom gain by being combined with other reduction techniques.

[1] Of course, in combination with other reduction techniques the optimal parameters for pruning might vary, but a brief trial showed that the original values as used here do not perform too badly.

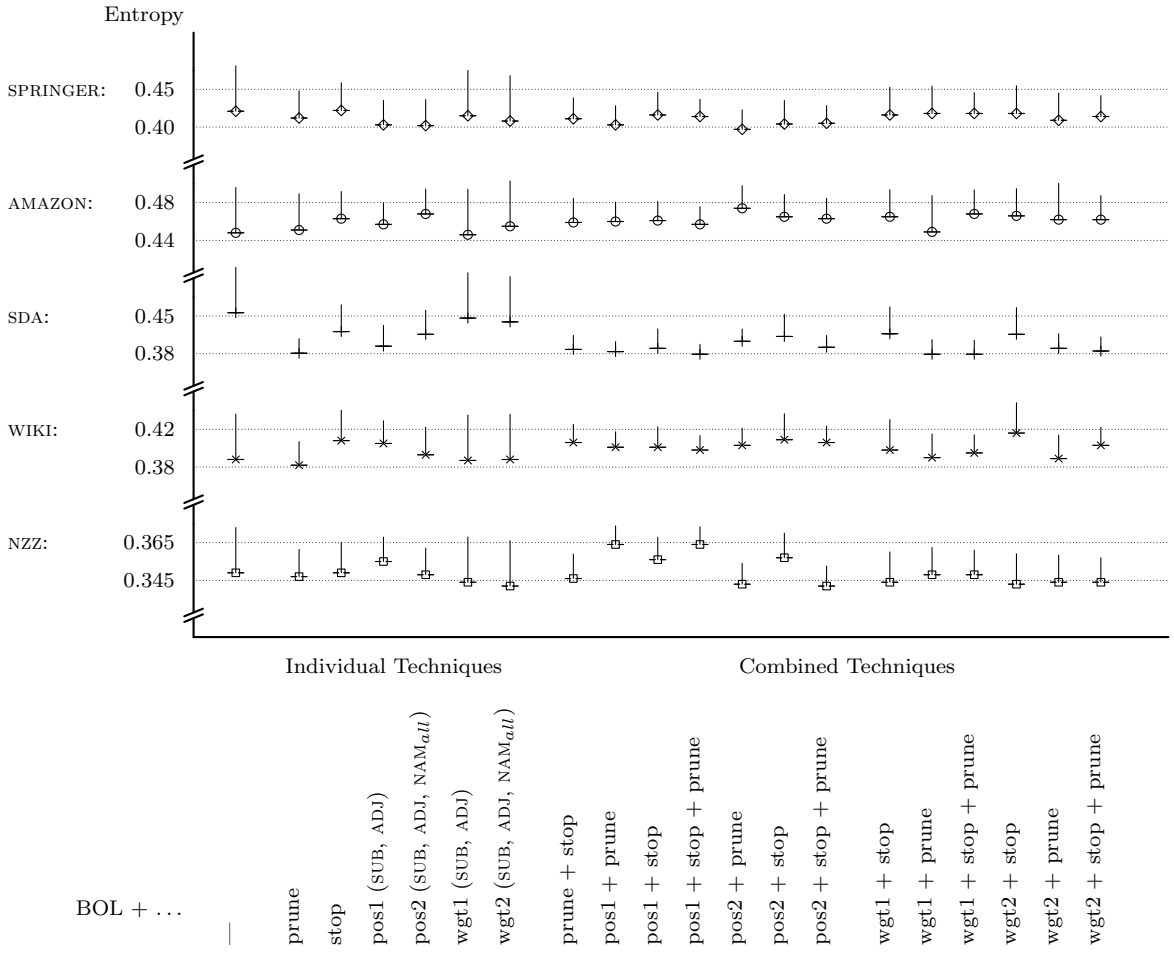


Figure 7.1: **Combining reduction techniques.** (For the underlying numbers refer to Table D.33.)

7.2 Combining Matrix Enhancement Techniques

From among the representation enhancement methods (Chapter 6), we chose three for further combining:

- **Compound splitting**, though for simplicity's sake we omitted to make use of any of the further modifications discussed in Section 6.1.3;
- **Multi-part proper names**, with a weighting factor of two;
- **Noun phrases**, also with a weighting factor of two.

The results depicted in Figure 7.2 contain few surprises. Combining techniques that improved the BOL_{stop} results tended to lead to good or better results, while the inclusion of disadvantageous techniques also tended to have a negative impact on the combined results. In particular, compound splitting was quite often better off without the addition of names or NPs, even though on their own these may have been useful.

On the whole the results confirm our findings of Chapter 6 and they encourage combining enhancement techniques that are really successful. At the same time, the SPRINGER and SDA results caution against combining a very successful technique (compound splitting) with an only moderately successful technique (names resp. names and NPs). In these cases relying on the main technique (compound splitting) appears to promise better results.

Finally, the very bad result in the NZZ set arising from compound splitting combined with NPs ought to be singled out; it is all the more surprising as both techniques were quite successful on their own. A closer inspection of the individual experimental runs that result in this average entropy of 0.359 reveals that the ten individual results fall into two groups. In six cases, the "AUSL" documents, which are usually almost all grouped together, are split in two big groups, leading to a final entropy value of 0.367 (see the sample confusion matrix in Table 7.1). In the other four cases the usual picture emerges (with only "FEUI" split up and "INLA" and "ZURI" being grouped together as before) and values of 0.346 to 0.349. It must be left open to speculation why the clustering process is suddenly geared into that unfavourable direction as often as six times out of ten. If compound splitting and NP addition are considered individually, the phenomenon does not occur in a single case out of twenty.[2]

This tendency to produce outliers was also demonstrated in a different experiment (see Figure 7.3). Here sub-samples of various sizes (from 10 to 90 percent of each data set) were clustered. While the SPRINGER and AMAZON sets show a more or less consistent entropy reduction with larger data sets and SDA and WIKI show a relatively stable behaviour (with a slight tendency in the opposite direction), the NZZ series is by far the least consistent, with various up and down turns. A closer inspection shows that many individual cluster results show either a "split" of the FEUI (Feuilleton) or AUSL (Ausland) clusters, while INLA (Inland) and ZURI (Zürich) regularly join each other.

[2]The same anomaly occurred in the later combinational experiments reported in Figure 7.4/Table D.35, by the way.

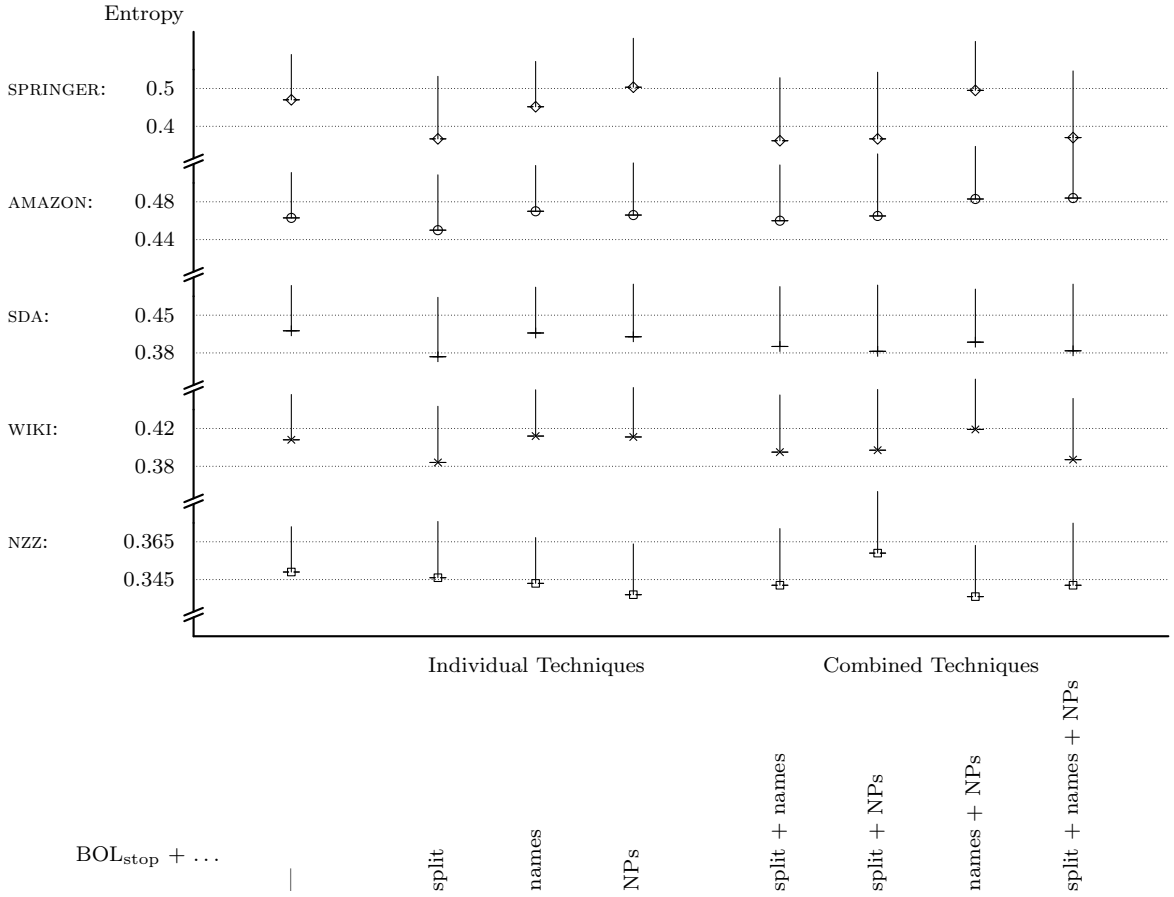


Figure 7.2: **Combining matrix enhancement techniques.** (For the underlying numbers refer to Table D.34.)

	FEUI	AUSL	INLA	SPOR	ZURI	VERM	WIRT
C_1	6842	92	106	12	374	400	22
C_2	156	4779	92	4	54	724	130
C_3	64	3183	144	1	5	45	13
C_4	45	55	2986	28	1861	223	257
C_5	4	0	0	3153	24	47	0
C_6	1297	511	274	30	540	3906	21
C_7	12	62	143	1	41	35	3063

Table 7.1: NZZ **confusion matrix with the category "AUSL" being split** (into clusters C_2 and C_3), leading to a significant entropy increase to 0.367.

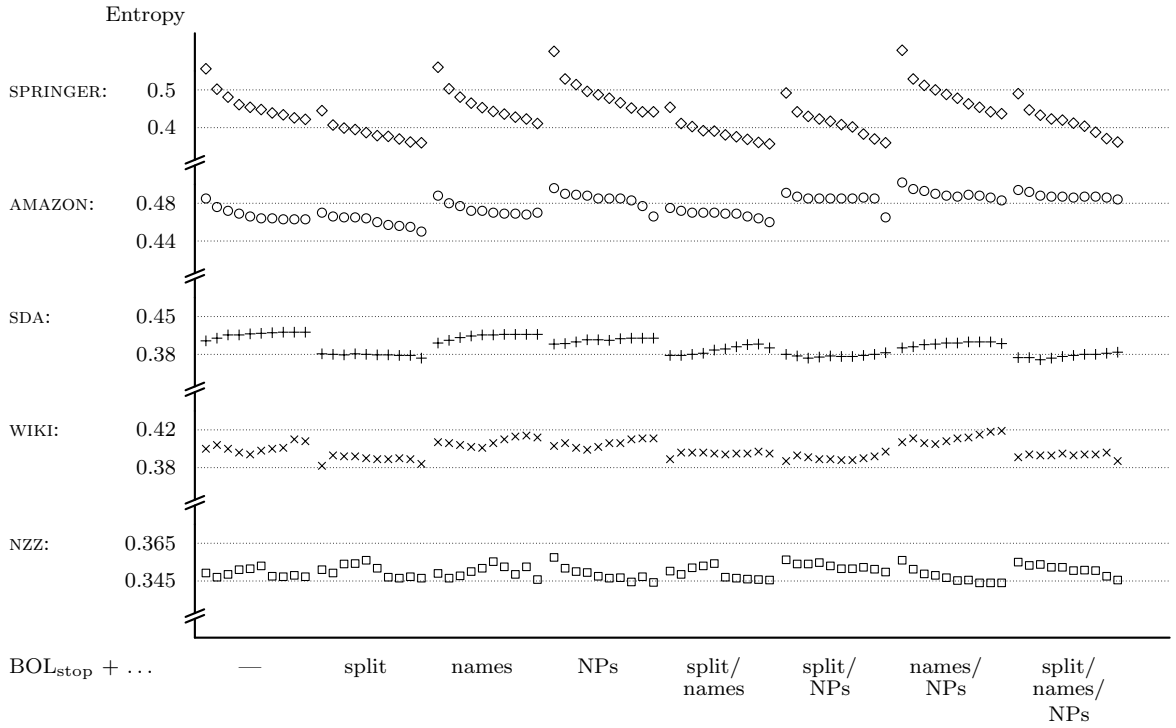


Figure 7.3: **Sub-sampling of matrix enhancement techniques.** For each method ten sub-samples of 10, 20 ... 80, 90 percent of all documents were taken and clustered. The ten data points per data set and method shown in the diagram are averaged entropy values for all nine sub-sample sizes (10 to 90%), with the full set value (100%) added on the right end of each mini-series. Only SPRINGER and AMAZON show a consistent improvement in clustering quality with a growing amount of data available.

7.3 Combining Matrix Reduction and Enhancement Techniques

In the third step of our combination experiment, we tried to combine techniques that have been identified as possibly useful from the previous two sections, namely: **pruning**, **stopword removal**, **POS feature weighting**, **compound splitting** and **noun phrase identification**. Figure 7.4 reports on the outcome.

By a comparison with the individual results from individual use and prior combinations of the various techniques, we can draw a number of tentative conclusions:

Pruning is able to reduce matrix size considerably, but it often leads to a deterioration of the clustering results. The pruning parameters, which were chosen individually for each data set (cf. Section 7.2), seem to be very sensitive to the actual circumstances and even within the same data set good parameters are not easily transferable. Deciding on which pruning parameters to use is therefore most difficult.

Stopword removal has here been shown to be a more reliable means of matrix reduction. In combination with the other techniques stopword removal tended to improve results for SPRINGER, SDA and WIKI. For AMAZON and NZZ it seems at least better than the pruning technique, even though the quality of the results is not greatly influenced by stopword removal.

POS weighting, which had been promising on several occasions when used on its own, did not combine so well with the other techniques and the SDA, WIKI and NZZ results were better without.

Noun phrases (NPs) had been a successful addition for the NZZ data set, but did not profit from combinations with other techniques. In general, the addition of NPs did not have too much of an extra impact. Its positive effects could still be felt in the SDA experiments. However, on the NZZ set the combination of NPs and compound splitting unexpectedly induced the clustering algorithm to produce a much inferior clustering with great regularity (in fact, the same effect occurred that has been discussed in the previous section).

Compound splitting was found essential for SPRINGER and generally useful for SDA and WIKI data. For AMAZON it was relatively beneficial only if stopword removal had been performed, while for NZZ it was less successful in combination with other techniques than they were on their own.

It is also noteworthy that compound splitting profits relatively often from prior stopword removal.

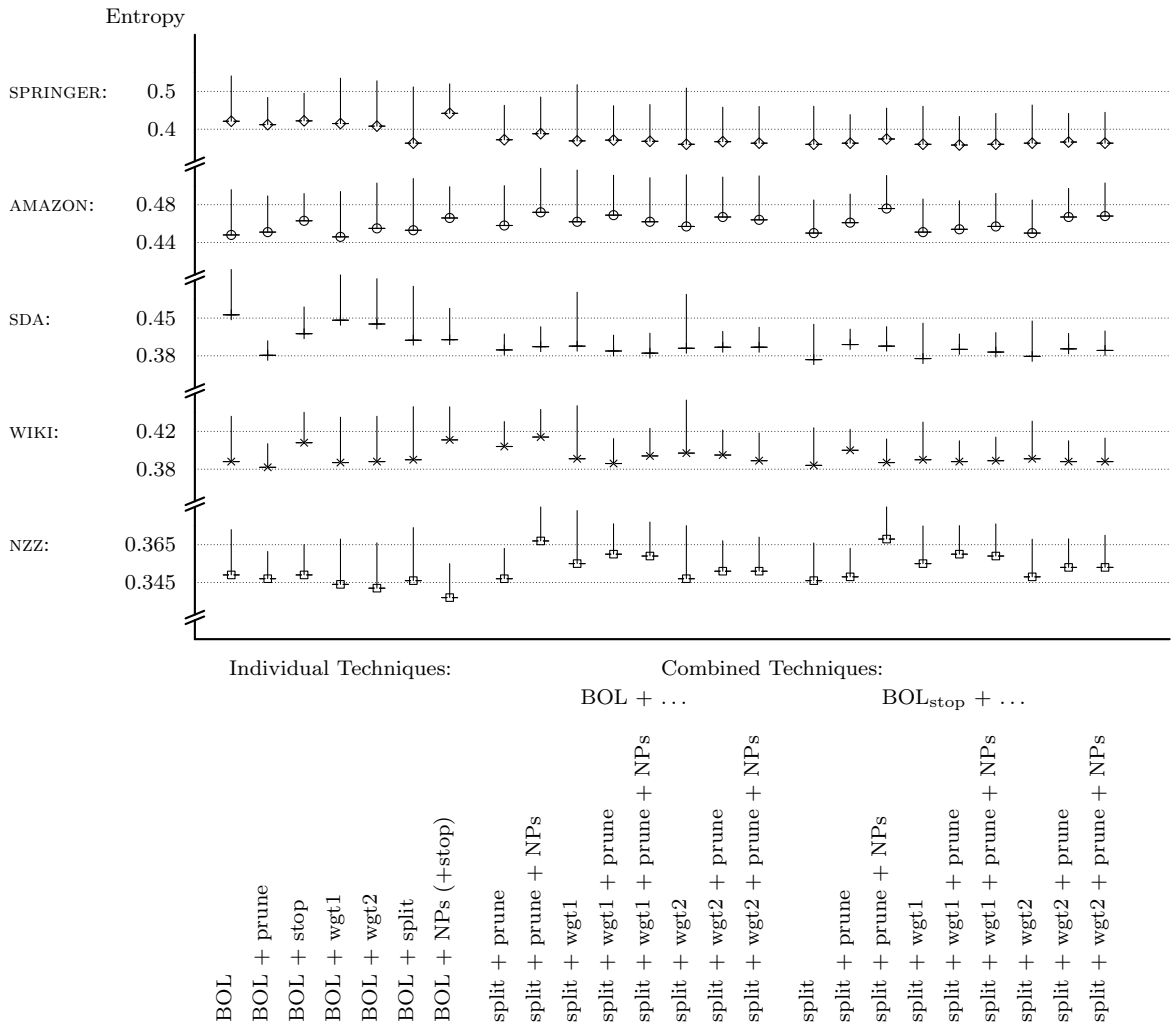


Figure 7.4: **Combining matrix enhancement and reduction techniques.** (For the underlying numbers refer to Table D.35.)

7.4 Conclusions

Our experiments have shown that on the whole the effects of combining different techniques can be roughly gauged by looking at the individual results, though not always. In particular, we have found a case (compound splitting and NPs for NZZ) where a significant deterioration was caused by the combination of two individually successful techniques. Combinations thus contain a risk of evoking unwanted side-effects and it would therefore seem advisable not to combine too many techniques.

Of the main two matrix reduction techniques we have found stopword removal to be more reliable than pruning since the latter depends strongly on the choice of suitable parameters. These parameters may change if several techniques are combined and it is thus practically very difficult to exhaust the potential that has previously been shown to be inherent to straight pruning.

Of the enhancement techniques, the usefulness of compound splitting has been confirmed. Moreover, it seems recommendable to combine it with stopword removal. Additional multi-word features such as phrases and names did not have much impact in the combination experiments.

Tentatively and based on the results of this and earlier chapters, we can conclude that these document representation techniques were most promising (under a combined qualitative/quantitative view):

- SPRINGER: stopword removal and compound splitting, (POS selection);
- AMAZON: stopword removal and compound splitting;
- SDA: stopword removal and compound splitting, (NPs);
- WIKI: stopword removal and compound splitting;
- NZZ: (stopword removal), POS weighting, NPs, names.

As an additional illustration, we combined these actions and chose the individually "optimal" parameters (as found earlier) for each data set.[3]

The results (Table 7.2) correspond more or less to our expectations. Again we find, however, that the combination of two "good" techniques does not automatically improve clustering results. In particular, compound splitting is turning out counter-productive if combined with the best possible stopword removal results for both AMAZON and WIKI (as opposed to the case with the standard stopword set, where splitting had proved useful).

For three of the five data sets (SPRINGER, SDA, NZZ) clear improvements over the baselines can be shown by using linguistic methods. For the other two, the improvements remained smaller (relatively speakingn) and—given that methods and parameters were selected *ex post*—not too conclusive.

[3]The parameters were chosen as follows:

	SPRINGER	AMAZON	SDA	WIKI	NZZ
stop (extracted)	E'':U-1000	WKL:U-100	E'':U-1000	E'':U-100 / df:U-50	E:U-1000
split	df 15	df 200	df 150	normal	—
POS selection	SUB, ADJ, NAM_{all}, UNK	—	—	—	—
POS weighting	—	—	—	—	2×: SUB, ADJ, NAM_{all}
Names (weight)	—	—	—	—	2×
NPs (weight)	—	—	2×	—	2×

SPRINGER	Baseline 1	0.410[0.010]
	Baseline 2	0.408[0.004]
	BOL	0.421[0.015]
	BOL_{stop}	0.422[0.018]
	stop*	0.407[0.020]
	stop* + split*	**0.355[0.003]**
	stop* + split* + pos*	0.399[0.019]
	stop* + pos* + split*	0.360[0.011]
AMAZON	Baseline 1	0.468[0.006]
	Baseline 2	0.458[0.009]
	BOL	0.448[0.007]
	BOL_{stop}	0.463[0.005]
	stop*	**0.446[0.007]**
	stop* + split*	0.458[0.007]
SDA	Baseline 1	0.431[0.002]
	Baseline 2	0.416[0.019]
	BOL	0.456[0.001]
	BOL_{stop}	0.421[0.001]
	stop*	0.373[0.008]
	stop* + split*	**0.372[0.009]**
	stop* + split* + NPs*	0.373[0.013]
WIKI	Baseline 1	0.416[0.019]
	Baseline 2	0.386[0.010]
	BOL	0.388[0.007]
	BOL_{stop}	0.408[0.019]
	stop1*	0.384[0.011]
	stop1* + split*	0.389[0.006]
	stop2*	**0.381[0.007]**
	stop2* + split*	0.402[0.012]
NZZ	Baseline 1	0.348[0.002]
	Baseline 2	0.345[0.001]
	BOL	0.349[0.005]
	BOL_{stop}	0.349[0.005]
	stop*	0.344[0.001]
	stop* + wgt*	0.340[0.004]
	stop* + wgt* + NPs	0.339[0.003]
	stop* + wgt* + NPs + names	**0.338[0.003]**

Table 7.2: **Results with "optimal" *(ex-post)* document representation.** Stars indicate the use of individually chosen parameters as listed in footnote 3 on page 181.

Chapter 8

Summary and Outlook

Harp not on that: nor do not banish reason
For inequality; but let your reason serve
To make the truth appear where it seems hid
And hide the false seems true.

William Shakespeare (*Measure for Measure*, ca. 1604)

In this final chapter we present a brief résumé of our experimental findings, putting them into relation with the different clustering scenarios introduced in Chapter 3.5 and trying to form conclusions and recommendations therefrom (Section 8.1). We then move on to a discussion of our main hypothesis and the benefits gained from natural language processing (Section 8.2), before concluding with a short outlook into future research areas (Section 8.3).

8.1 Document Representation Techniques for Clustering

In the following few paragraphs we try to summarise the experimental findings of this study. Our main goal was to establish the usefulness of natural language processing (NLP) for a particular task, i. e. clustering German-language documents. Inevitably, though, our experiments led us to examine a number of non-linguistic issues as well, with sometimes interesting consequences.

In Section 3.5 we had introduced a model of four different clustering scenarios: off-line, repeated, ad-hoc and instant clustering, each of which operates with its own specific time and memory constraints. A proper assessment of document representation techniques needs to take these aspects into account, in addition to clustering quality.

IDF squared. Our first major discovery (Section 4.4.2) concerns the overall (global) weighting scheme. We found that the customary IDF scheme performed markedly worse than the squared variant IDF^2 on at least four of the five German data sets. IDF^2 has so far received only very little attention in the information retrieval literature but following this study probably deserves to be investigated more thoroughly. The large step forward in clustering quality and the small cost involved suggest that this method should be strongly considered in all clustering applications, even if matrix assembly is time-critical as in ad-hoc and instant clustering.

Lemmatising. Unless motivated by additional processing steps later on (which require lemmata as input), we have not found evidence that lemmatising justifies the considerable extra effort required in comparison with simple, crude stemming (Section 5.2). For applications with a time-critical document vectorisation stage (instant clustering), lemmatising is thus not recommended.

Pruning. We observed that in some (but not all) cases "pruning" (removal of information based only on statistics, i. e. occurrence frequencies) is not only a means of reducing the complexity of the clustering task but can also improve cluster results (Section 5.3). In fact, in some cases it was possible to remove up to 80% of the data (non-zero elements) without seriously affecting the outcome of the clustering algorithm.

It also transpired, however, that the effects of pruning are differing widely for the individual data sets. In particular, it is difficult to determine the cut-off points for pruning; they seem to vary strongly as well. Generally speaking, the results gave the impression that for tasks with few clusters, a generous global pruning strategy (based on document frequencies) was applicable, whereas for the more delicate tasks (with many clusters) a local pruning strategy (individually for each document) seemed preferable. The number of data sets was insufficient, however, to draw firm conclusions.

Stopword removal. Removing stopwords from documents is an old technique requiring no linguistic skills either (though *creating* the stopword list may do). Our combination experiments (Chapter 7) showed that stopword removal was a more reliable way of reducing matrix size than pruning. On the other hand, quite contrary to intuition and general consensus, we had to note that stopword removal was *not* universally recommendable. On the contrary, for two of our five data sets it seems that stopword removal hampered the clustering process (Section 5.4).

We further discussed the possibility that the particular stopword list, which is comparatively large, may have been responsible for the result, but found that this was not so. We further examined a number of statistical techniques to create stopword lists from labelled document collections. It emerged that using appropriate discrimination measures and parameters, it is quite possible to come up with suitable stopword lists by an automated process. However, the phenomenon of certain cluster results being virtually impossible to improve through stopword removal persisted.

Based on our experiments, it is difficult to give a definite recommendation with regard to stopword removal. It remains a prime tool for quick and simple matrix size reduction at vectorisation stage, with relatively little risk of going badly astray. Nevertheless, it has been shown that so-called stopwords should not be removed mindlessly and had sometimes better be retained. Furthermore we found evidence that the stopword question becomes more important with short documents than with longer ones.

POS selection. Selecting features based on their POS tags revealed that nouns and adjectives were the most important words followed by proper names, while verbs had little impact (Section 5.5). The qualitative consequences of POS restriction varied between the data sets. In comparison with pruning it was found that POS selection was relatively reliable, leading to better results than the pruning methods with comparable reduction factors.

From a qualitative point of view it is not possible to recommend POS selection for all situations. From a quantitative point of view, however, it is a promising means of reducing matrix size considerably (between 40 and 50 percent), with little risk of losing vital informa-

tion. Like stopword removal, POS selection can already be performed at the vectorisation stage and thus is a good candidate for fastening ad-hoc clustering systems.

POS/stopword weighting. As an alternative to removal of stopwords and "stop POS tags" we examined the possibility of down-weighting them (Section 5.7.1). Of course, this precludes any gains in speed or memory requirements. Since the qualitative improvements were also questionable, this smoothing alternative to removal was found unsatisfactory.

Compound splitting. One of the most interesting investigations concerned the morphological analysis of lemmata and subsequent splitting of compounds into their constituents. Since German is very rich in compounds (30–45% of all word types in our data), the effect would be expected to be considerable. In order to avoid too many new and irrelevant features, it was found useful to combine it with stopword removal. Compound splitting then proved indeed a very powerful tool, improving the results for the majority of data sets by a large margin (Section 6.1).

Yet, the success was not as uniform as might have been expected. With two sets, the addition of compound splitting was only successful if stopword removal had been performed before (Section 7.3). However, stopword removal having proven disadvantageous for these same two sets in the first place, the combined effect of stopword removal and compound splitting was only about enough to cancel the disadvantage of using stopword removal (nor was compound splitting without stopword removal any better).

Thus, our data only partially supports the opinion that for a productive language such as German compound splitting is absolutely compulsory since the effect cannot be separated entirely from stopword removal. Our attempts to refine compound splitting (to cases where it was *really* useful) were not crowned with much success. We only found certain evidence that it might pay off to leave highly frequent compound terms intact. We also found that in our experiments (unlike in those of Rosell, 2003) it was advisable to keep the compounds as features even after splitting.

The morphological analysis necessary for compound splitting can be performed at the vectorisation stage, which means that it is suitable for all clustering types except instant clustering.

Multi-part Names and NPs. Using syntactic information to create composite features (names and noun phrases) has brought only partial success (Section 6.2). The dividends were biggest with the NZZ data set. As this happened to be by far the one with the longest texts (and perhaps richest language), we hypothesize that there is a connection between the richness of a document and the benefits that can be gained by calculating such extra features. Nevertheless, the overall results left some doubts as to whether the extra effort was justified. Further investigations will be necessary. These features can also be calculated at vectorisation stage. It should be kept in mind, however, that in particular syntactic parsing (necessary for NLP) is very time-consuming and storing all phrases for each document increases memory requirements by a substantial factor.

Semantics. Experiments using a German ontology (GermaNet) could not be brought to a success (Section 6.3). In our opinion this failure is not so much due to the quality of the ontology or its extent, but mainly due to the difficulty of distinguishing between concept-relations that were succinct and relevant to our specific purposes, and relations that were too general. On top of this comes the word sense disambiguation problem, though it may be secondary to the problem of selecting just the "good" parts of the ontology.

Based on our experiments, we cannot recommend the use of a general ontology without important methodological improvements.

Cluster granularity. We have observed in various instances that refinement and pruning techniques that were successful for the 5- and 7-class data sets had no or a negative impact on the 21- and 22-class data. We observed that the same data sets behaved quite "normally" when only subsets of fewer classes were considered (cf. Sections 5.6 and 6.1.4). Our hypothesis from these (few) observations is that "crude" clustering tasks with relatively few clusters are more likely to profit from document representation refinement than the more complex tasks. In the latter case the equilibrium appears to be more sensitive to changes, and the benefits gained in one area of the clustering landscape are more likely to be weighed up by erroneous re-adjustments of cluster frontiers in another area. So far, this is only a working hypothesis based on a very small number of samples and future dedicated experiments are necessary to throw more light on the phenomenon.

8.2 The Case of Natural Language Processing

The primary goal of this study was to test the hypothesis that natural language processing (NLP) tools are able to improve the input to document clustering algorithms. Perusal of our findings in the preceding section reveals that there can be no simple answer to this question.

For various of the techniques under examination we found partial, sometimes very strong evidence in favour of NLP. But we also noted a significant number of exceptions, and so the decision will have to depend on the available resources and the concrete goals of a clustering algorithm.

Of the NLP applications we have found *compound splitting* (necessitating proper morphological analysis and lemmatising) as well as *noun phrase extraction* (in suitably rich corpora) to be serious candidates for improving clustering quality. For complexity reduction we have found *part-of-speech filtering* to be a reliable means of excluding much superfluous information.

Compared to the actual clustering process, the document preparation step is time-consuming, in particular if it involves NLP. For instance, morphological analysis with GERTWOL took almost 20 times longer for the SPRINGER set than clustering the SPRINGER data. For instant clustering purposes, NLP techniques are thus hardly an option unless the number of documents is very small. With most of NLP taking place at the vectorisation stage, it is, however, equally well-suited for off-line, repeated and ad-hoc clustering. Given texts of sufficient length, the use of the above-mentioned NLP techniques can thus be recommended.

Finally, it should be noted that the main focus of our experiments was on clustering quality, with a secondary focus on matrix size (which is related to clustering speed as well as storage requirements). We did not, however, include the aspect of cluster description ("cluster digest") in our investigations at all. Such considerations will no doubt move the balance somewhat in favour of the linguistic methods which offer additional options to come up with humanly understandable, "good" cluster descriptions.

8.3 Future Research

It could be shown in the present study with five relatively large data sets that linguistic methods bear the potential for qualitative improvements for a typical number-crunching application such as document clustering. On the other hand, it was also shown that there was no guarantee for such improvements. One goal of future research must be to investigate more precisely the

circumstances that favour the use of linguistic tools. Our hypothesis from the present experiment is that length and richness of the texts as well as the number of clusters play a role, but the evidence needs to be broadened. We had to do with five data sets in this study; in order to gain statistically reliable and significant results 30 or more data sets of similar size would be desirable.

On the technical level we found a more thorough investigation of ontology usage necessary and possibly quite rewarding. We have been unable in that context to come up with a suitable algorithm for separating useful from useless information in this study, but still feel that improvements should be possible—even for unstructured and basically unlimited application areas (as was the case with our five data sets).

The research into the nature of stopwords has been shown to be a topic deserving additional investigations. In particular, it would be desirable to have more knowledge about situations where stopword removal can be *harmful*, a possibility hitherto often neglected. Besides, it would be desirable to subject our automatic stopword identification techniques to larger scale evaluations and for different IR applications. The same applies to the IDF^2 weighting method. It needs further practical evaluations, but has made a very promising start.

As regards document clustering itself, the choice of document representation techniques, clustering algorithm and cluster description method will often depend on the final purpose. More research is still needed to identify, classify and characterise typical applications. As a start, we have put forth a scheme of four typical scenarios (from off-line to instant clustering). It is also worthwhile to investigate the potential of clustering applications in a more global research and information seeking context, from which new requirements and adaptions of existing algorithms may be derived. In particular, we see a promising area in the exploration of "dynamic" clustering algorithms which allow the user to interact actively with the clustering system and to take direct influence on the clustering results; for instance, by tuning certain weighting functions (e. g. for proper names), defining "stopwords" on-the-fly or manipulating the cluster structure by causing the system to merge, split or dissolve certain clusters.

Overall, there can be little doubt that as a means of accessing large and as yet unstructured bodies of textual data, document clustering will remain a promising instrument in the information scientist's toolbox.

Appendix A

Glossary

Terms within double-quotes were introduced in the context of the present thesis.

Bahuvrihi. An exocentric compound, i. e. a compound whose meaning is not contained in any of the constituents (e. g. redskin).

BIRCH. *Balanced Iterative Reducing and Clustering using Hierarchies.* A single-pass hierarchical clustering algorithms for large data sets (Zhang *et al.*, 1996).

BOL. *Bag-Of-Lemmata.* An alternative to the bag-of-words model (→BOW), based on lemmata instead of word forms.

BOW. *Bag-Of-Words.* A document model which represents a document by the sum of its word tokens without preserving the initial word order.

Categorisation/Classification. The task of assigning similar objects to predefined categories/groups.

CLARA. *Clustering LArge Applications.* A medoid-based clustering algorithm.

CLARANS. *Clustering Large Applications based on RANge Search.* Another medoid-based clustering algorithm.

Cluster Analysis/Clustering. The task of grouping similar objects together based on their properties without *a priori* assumptions about the evolving groups.

Cluto. *CLUstering TOolkit.* A powerful clustering software developed at the University of Minnesota (Karypis, 2003).

Compound. A word made up from two or more other words.

CURE. *Clustering Using REpresentatives.* A clustering algorithm suggested by Guha *et al.* (1998).

Data Mining. The task of extracting information from large bodies of data with automated tools.

Document Frequency. The document frequency of a term is the number of documents in a collection that share that term.

EM. *Expectation-Maximisation.* A probabilistic maximum-likelihood algorithm for clustering (Dempster *et al.*, 1977).

Fugenlaut. A letter working as "glue" in a German compound (e. g. the letter "n" in "Fuge-n-laut").

GermaNet. A German ontology built on the same principles than →WordNet.

HAC. *Hierarchical Agglomerative Clustering.* The class of clustering algorithms building a bottom-up hierarchy of clusters.

HDC. *Hierarchical Divisive Clustering.* The class of clustering algorithms building a top-down hierarchy of clusters.

HTML. *HyperText Markup Language.* A set of formatting commands used for documents on the Internet.

Hyperlink. An interactive reference between documents on the →WWW.

Hypernym. A superordinate word for a given word.

Hyponym. A subordinate word for a given word.

IR. *Information Retrieval.* The discipline concerned with the storage and retrieval of information (most often in the form of documents).

ISBN. *International Standard Book Numbering.* A system assigning unique identifiers to books.

LSA. *Latent Semantic Analysis.* See →LSI.

LSI. *Latent Semantic Indexing.* An algebraic matrix complexity and noise reduction technique using →SVD. It does not contain any semantics in a narrower sense, but aims to reduce a feature matrix to the principal "concepts".

LZW Algorithm. A data compression algorithm introduced by Ziv and Lempel (1977) and Welch (1984).

"Mechanical Compound." A →compound that can be recognised without linguistic analysis as it combines the constituents by a hyphen or a slash.

Meta Search Engine. Search engine that does not use a data repository (index) of its own, but gathers search results from multiple other search engines and presents an integrated view of the combined results.

NLP. *Natural Language Processing.* Summaric term for all algorithms, models, data and theoretical background used for the automated analysis of electronic textual data involving knowledge about natural languages.

OHSUMED. A large database of labelled medical abstracts often used in →IR.

"Organic Compound." A →compound that can only be recognised and split up with the help of a morphological analysis.

PAM. *Partitioning Around Medoids.* A medoid-based clustering algorithm (cf. Kaufman and Rousseeuw, 1990).

PDDP. *Principal Direction Divisive Partitioning.* An efficient clustering algorithm for constructing a cluster hierarchy "top-down" (Moore *et al.*, 1997).

PDF. *Portable Document Format.* A popular document storage format ensuring an identical, platform-independent rendering of the content.

POS. *Part-Of-Speech.* The grammatical function of a word in a sentence.

PostScript. A popular document format for printing purposes.

ROCK. A clustering algorithm for categorical data as opposed to numeric data (Guha *et al.*, 2003).

SGML. *Standard Generalized Markup Language.* A meta-language for document annotation.

"Shared Feature." A feature that occurs in more than one of the documents to be clustered. The opposite is a →"unique feature".

Snippet. A small preview of a document in a search engine results page—usually one or two "most relevant" passages from the document.

SOM. *Self-Organising Map.* A classification technology derived from neural network research (a sub-domain of Artificial Intelligence).

Stemming. The process of truncating words at the end by removing letters and transforming them according to a few pre-defined, language-dependent rules (but without recourse to a lexicon).

Stopword. A functional word bearing little or no actual content.

"Stopwordliness." The degree to which a term acts as a →stopword.

SVD. *Singular Value Decomposition.* An algebraic technique very similar to principal component analysis, used to reduce a matrix to its main components.

Synonym. A word with an (almost) identical meaning as another given word.

Tagging. The process of annotating a text with meta-data, for instance with →POS tags.

Text Data Mining. The sub-discipline of →data mining concerned with automatically extracting information from large bodies of textual data.

Tokenisation. The process or recognising word boundaries and splitting a text string into a sequence of individual entities (usually words).

TREC. *Text REtrieval Conference.* An annual international conference for text and information retrieval, organised by the American NIST (National Institute of Standards and Technology).

Truncation. The process of truncating words at the end by removing a fixed number of letters.

"Unique Feature." A feature that occurs in only one of the documents to be clustered. The opposite is a →"shared feature".

UNL. *Universal Networking Language.* An ambitious project aiming at a universal semantic representation of concepts and texts.

UPGMA. *Unweighted Pairwise Group Method with Averages.* The most popular →HAC criterion.

UPGMC. *Unweighted Pairwise Group Method with Centroids.* An occasionally used →HAC criterion.

URL. *Uniform Resource Locator.* A unique and universal address for a document or resource on the →WWW.

Vectorisation. The process of analysing a document and mapping it to a feature vector.

Web. Short for World Wide Web (→WWW).

WordNet. A popular general ontology for English introduced by Miller *et al.* (1990).

WPGMA. *Weighted Pairwise Group Method with Averages.* A rarely used →HAC criterion.

WPGMC. *Weighted Pairwise Group Method with Centroids.* An →HAC criterion with little practical importance.

WWW. *World Wide Web.* The entire collection of static and dynamic documents and data linked together on the Internet.

XML. *EXtensible Markup Language.* A popular annotation language derived from →SGML.

Appendix B

Proofs

The proofs given in this Appendix partially overlap and refine those given by Zhao and Karypis (2001).

B.1 Equivalence of Euclidean and Cosine Similarity

Here follows the proof of the equivalence of cosine similarity and Euclidean similarity, provided that the document vectors all have unit length. Equation 2.11 gave the following formula, which shall now be proven:

$$\hat{s}_{Euclid}(\mathbf{d}_i, \mathbf{d}_j) = \sqrt{2 - 2s_{Cosine}(\mathbf{d}_i, \mathbf{d}_j)}, \quad \text{with } \|\mathbf{d}_i\|_2 = \|\mathbf{d}_j\|_2 = 1. \tag{B.1}$$

By definition (2.4)

$$\hat{s}_{Euclid}(\mathbf{d}_i, \mathbf{d}_j) = \sqrt{\sum_{k=1}^{m}(d_{ik} - d_{jk})^2} \tag{B.2}$$

$$= \sqrt{\sum_{k=1}^{m} {d_{ik}}^2 + \sum_{k=1}^{m} {d_{jk}}^2 - 2\sum_{k=1}^{m} d_{ik}d_{jk}}, \tag{B.3}$$

and because of the document vectors having unit length, this can be simplified to

$$\hat{s}_{Euclid}(\mathbf{d}_i, \mathbf{d}_j) = \sqrt{2 - 2\cos(\mathbf{d}_i, \mathbf{d}_j)}. \tag{B.4}$$

□

B.2 Minimum Variance Simplification

In order to prove Equation 2.23, let $\mathbf{r}^{\mathrm{c}}_{1\cup 2}$ be the centroid of the joint cluster $C_1 \cup C_2$. Since

$$\sum_{\mathbf{d}_j \in C_1} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_{1\cup 2}\|_2^2 = \sum_{\mathbf{d}_j \in C_1} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_1\|_2^2 + n_1 \|\mathbf{r}^{\mathrm{c}}_1 - \mathbf{r}^{\mathrm{c}}_{1\cup 2}\|_2^2 \,, \tag{B.5}$$

we can rewrite Equation 2.21 as follows

$$\begin{aligned}
\hat{S}(C_1, C_2) &= \mathrm{ESS}(C_1 \cup C_2) - \mathrm{ESS}(C_1) - \mathrm{ESS}(C_2) && \text{(B.6)}\\
&= n_1 \|\mathbf{r}^{\mathrm{c}}_1 - \mathbf{r}^{\mathrm{c}}_{1\cup 2}\|_2^2 + \sum_{\mathbf{d}_j \in C_1} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_1\|_2^2 \\
&\quad + n_2 \|\mathbf{r}^{\mathrm{c}}_{1\cup 2} - \mathbf{r}^{\mathrm{c}}_2\|_2^2 + \sum_{\mathbf{d}_j \in C_2} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_2\|_2^2 \\
&\quad - \sum_{\mathbf{d}_j \in C_1} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_1\|_2^2 \\
&\quad - \sum_{\mathbf{d}_j \in C_2} \|\mathbf{d}_j - \mathbf{r}^{\mathrm{c}}_2\|_2^2 && \text{(B.7)}\\
&= n_1 \|\mathbf{r}^{\mathrm{c}}_1 - \mathbf{r}^{\mathrm{c}}_{1\cup 2}\|_2^2 + n_2 \|\mathbf{r}^{\mathrm{c}}_{1\cup 2} - \mathbf{r}^{\mathrm{c}}_2\|_2^2 \,. && \text{(B.8)}
\end{aligned}$$

Since

$$\mathbf{r}^{\mathrm{c}}_{1\cup 2} = \frac{n_1 \mathbf{r}^{\mathrm{c}}_1 + n_2 \mathbf{r}^{\mathrm{c}}_2}{n_1 + n_2}, \tag{B.9}$$

Equation B.8 can be further simplified to

$$\hat{S}(C_1, C_2) = \frac{\|\mathbf{r}^c_1 - \mathbf{r}^c_2\|_2^2}{\frac{1}{n_1} + \frac{1}{n_2}} \,. \tag{B.10}$$

□

B.3 Comparative Examination of Internal Cluster Criteria

In this section we examine the different *internal clustering criteria* (see Section 2.3.1.1) that can be expressed according to Equation 2.25 as

$$E(C_i) = w\frac{S}{a}. \tag{B.11}$$

Again we assume that all documents are normalised to unit length:

$$\forall \mathbf{d} \in \mathcal{D} : \|\mathbf{d}\| = 1 \,. \tag{B.12}$$

Furthermore, let $\mathbf{y}_i$ again be the composite vector of cluster C_i (Eq. 2.28):

$$\mathbf{y}_i = \sum_{\mathbf{d}_j \in C_i} \mathbf{d}_j \,, \tag{B.13}$$

with the length of the composite vector

$$\begin{aligned} \|\mathbf{y}\| &= \sqrt{\mathbf{y}^T\mathbf{y}} \\ &= \sqrt{\sum_{i=1}^{|C|}\sum_{j=1}^{|C|}\mathbf{d}_i{}^T\mathbf{d}_j}. \end{aligned} \tag{B.14}$$

We then examine the function $\Psi(\mathcal{C}) = \sum E(C_i)$ for the following different choices of S:

- Case A: $\sum\sum s_{Cosine}(\mathbf{d}_i, \mathbf{d}_j)$,
- Case B: $\sum\sum s_{Euclid}(\mathbf{d}_i, \mathbf{d}_j)^2$,
- Case C: $\sum s_{Cosine}(\mathbf{d}, \mathbf{r}^{\mathrm{c}})$,
- Case D: $\sum s_{Cosine}(\mathbf{d}, \mathbf{r}^{\hat{\mathrm{c}}})$.
- Case E: $\sum s_{Euclid}(\mathbf{d}, \mathbf{r}^{\mathrm{c}})^2$,
- Case F: $\sum s_{Euclid}(\mathbf{d}, \mathbf{r}^{\hat{\mathrm{c}}})^2$.

(For the definitions of $\mathbf{r}^{\mathrm{c}}$ and $\mathbf{r}^{\hat{\mathrm{c}}}$ refer to Equations 2.13 and 2.14.)

Case A

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|}\sum_{j=1}^{|C|}\cos(\mathbf{d}_i, \mathbf{d}_j) \\ &= \sum_{i=1}^{|C|}\sum_{j=1}^{|C|}\mathbf{d}_i^T\mathbf{d}_j \\ &= \|\mathbf{y}\|^2. \end{aligned} \tag{B.15}$$

Case B

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|}\sum_{j=1}^{|C|}\|\mathbf{d}_i - \mathbf{d}_j\|^2 \\ &= \sum_{i=1}^{|C|}\sum_{j=1}^{|C|}(\mathbf{d}_i - \mathbf{d}_j)^T(\mathbf{d}_i - \mathbf{d}_j) \\ &= \sum_{i=1}^{|C|}\sum_{j=1}^{|C|}\mathbf{d}_i{}^T\mathbf{d}_i + \mathbf{d}_j{}^T\mathbf{d}_j - 2\mathbf{d}_i{}^T\mathbf{d}_j \\ &= 2|C|^2 - 2\|\mathbf{y}\|^2. \end{aligned} \tag{B.16}$$

Case C

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|} \cos(\mathbf{d}_i, \mathbf{r}^{\mathrm{c}}) \\ &= \sum_{i=1}^{|C|} \frac{{\mathbf{d}_i}^T \cdot \frac{\mathbf{y}}{|C|}}{\|\mathbf{d}_i\| \cdot \left\| \frac{\mathbf{y}}{|C|} \right\|} \\ &= \frac{\|\mathbf{y}\|^2/|C|}{\|\mathbf{y}\|/|C|} \\ &= \|\mathbf{y}\|. \end{aligned} \tag{B.17}$$

Case D

Since the cosine is not depending on vector length, this measure is obviously the same as that in case C:

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|} \cos(\mathbf{d}_i, \mathbf{r}^{\hat{\mathrm{c}}}) \\ &= \sum_{i=1}^{|C|} {\mathbf{d}_i}^T \cdot \frac{\mathbf{y}}{\|\mathbf{y}\|} \\ &= \|\mathbf{y}\|. \end{aligned} \tag{B.18}$$

Case E

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|} \|\mathbf{d}_i - \mathbf{r}^{\mathrm{c}}\|^2 \\ &= \sum_{i=1}^{|C|} {\mathbf{d}_i}^T \mathbf{d}_i + \mathbf{r}^{\mathrm{c}T} \mathbf{r}^{\hat{\mathrm{c}}} - 2 \cdot \mathbf{d}_i \mathbf{r}^{\hat{\mathrm{c}}} \\ &= |C| + |C| \cdot \left(\frac{\|\mathbf{y}\|}{|C|} \right)^2 - 2 \frac{\|\mathbf{y}\|^2}{|C|} \\ &= |C| - \frac{\|\mathbf{y}\|^2}{|C|}. \end{aligned} \tag{B.19}$$

Case F

$$\begin{aligned} S(C) &= \sum_{i=1}^{|C|} \|\mathbf{d}_i - \mathbf{r}^{\hat{\mathrm{c}}}\|^2 \\ &= \sum_{i=1}^{|C|} {\mathbf{d}_i}^T \mathbf{d}_i + \mathbf{r}^{\hat{\mathrm{c}}^T} \mathbf{r}^{\hat{\mathrm{c}}} - 2 \frac{{\mathbf{d}_i}^T \mathbf{y}}{\|\mathbf{y}\|} \\ &= 2|C| - 2\|\mathbf{y}\|. \end{aligned} \tag{B.20}$$

	$\mathbf{S}(C)$	$\mathbf{a}$	$\Psi(\mathcal{C}\|a,S,w)$ **unweighted** $(w=1)$	$\Psi(\mathcal{C}\|a,S,w)$ **weighted** $(w=\|C\|)$
A	$\sum\sum\cos(\mathbf{d}_i,\mathbf{d}_j)$	$2\|C\|^2$	$\max\sum^k\left(\|\mathbf{y}\|^2/2\|C\|^2\right)$	$\max\sum^k\left(\|\mathbf{y}\|^2/2\|C\|\right)$
B	$\sum\sum\|\mathbf{d}_i-\mathbf{d}_j\|^2$	$2\|C\|^2$	$\min\ k-\sum^k\left(\|\mathbf{y}\|^2/\|C\|^2\right)$	$\min\ n-\sum^k\left(\|\mathbf{y}\|^2/\|C\|\right)$
C/D	$\sum\cos(\mathbf{d},\mathbf{c})$	$\|C\|$	$\max\sum^k\left(\|\mathbf{y}\|/\|C\|\right)$	$\max\sum^k\|\mathbf{y}\|$
E	$\sum\|\mathbf{d}-\mathbf{r}^{\mathrm{c}}\|^2$	$\|C\|$	$\min\ k-\sum^k\left(\|\mathbf{y}\|^2/\|C\|^2\right)$	$\min\ n-\sum^k\left(\|\mathbf{y}\|^2/\|C\|\right)$
F	$\sum\|\mathbf{d}-\mathbf{r}^{\hat{\mathrm{c}}}\|^2$	$\|C\|$	$\min\ 2k-2\sum^k\left(\|\mathbf{y}\|/\|C\|\right)$	$\min\ 2n-2\sum^k\|\mathbf{y}\|$

Table B.1: **Summary of different internal clustering criteria.**

Summary

Table B.1 summarises the twelve criterion functions Ψ optimised for six different choices of S, each once in the weighted and once in the unweighted form. Evidently, cases B and E lead to the same result. In addition, minimising B or E leads to the same result as maximising A. Similarly, C, D and F are also equivalent, so that all in all there are effectively just four distinctly different criteria.

B.4 External Clustering Criteria

In this section we prove that the external Euclidean criterion $\max\sum\|\mathbf{r}^{\mathrm{c}}-\mathbf{c}_{\mathcal{S}}\|^2$ is equivalent to the internal Euclidean criterion $\min\sum\|\mathbf{d}-\mathbf{r}^{\mathrm{c}}\|^2$, as claimed towards the end of Section 2.3.1.1.

In the following proof let

$$\mathbf{c}_i=\mathbf{r}_i^{\mathrm{c}}\quad\text{and}\quad\mathbf{c}_{\mathcal{S}}=\mathbf{r}_{\mathcal{S}}^{\mathrm{c}}. \tag{B.21}$$

See also Equations 2.13 and 2.32.

$$\begin{aligned}
\Psi(\mathcal{C}) &= \sum_{i=1}^{k}|C_i|\cdot\|\mathbf{c}_i-\mathbf{c}_{\mathcal{S}}\|^2 && \text{(B.22)}\\
&= \sum_{i=1}^{k}|C_i|\left({\mathbf{c}_i}^T\mathbf{c}_i+{\mathbf{c}_{\mathcal{S}}}^T\mathbf{c}_{\mathcal{S}}-2{\mathbf{c}_i}^T\mathbf{c}_{\mathcal{S}}\right) && \text{(B.23)}\\
&= \sum_{i=1}^{k}|C_i|\left(\frac{{\mathbf{y}_i}^T\mathbf{y}_i}{|C_i|^2}+\frac{\left(\sum\mathbf{y}\right)^T\sum\mathbf{y}}{n^2}-2\frac{{\mathbf{y}_i}^T\sum\mathbf{y}}{|C_i|n}\right) && \text{(B.24)}\\
&= \sum_{i=1}^{k}\left(\frac{\|\mathbf{y}_i\|^2}{|C_i|}\right)+\frac{\|\mathbf{y}_{\mathcal{S}}\|^2}{n}-\frac{2}{n}\sum\|\mathbf{y}_{\mathcal{S}}\|^2 && \text{(B.25)}
\end{aligned}$$

$$= \sum_{i=1}^{k} \left(\frac{\|\mathbf{y}_i\|^2}{|C_i|} \right) - \frac{\|\mathbf{y}_{\mathcal{S}}\|^2}{n}. \tag{B.26}$$

Since the second term in Equation B.26 is a constant, maximising this last equation leads to the same result as minimising the weighted function for case E in Table B.1, i. e. $n - \sum \left(\|\mathbf{y}\|^2 / |C| \right)$.

□

Appendix C

Stoplists

C.1 German

ab
abend
aber
abgehandelt
abseits
absolut
abzueglich
acht
achtziger
aehnlich
aeusser
aeussern
aeusserst
all
alle
allein
allem
allen
allenfalls
aller
allerdings
allerlei
alles
allesamt
allgemeinen
allmaehlich
allzu
allzuviel
als
also
am
an
an-das
an-der
ander
andere
anderem
anderen
anderer
andererseits
anderes
anderlei
anderm
andermal
andern
andernorts
anders
anderseits
anderswo
andre
andrem
andren
andrer
andres
anfangs
angeben
angeblich
angesichts
anhand
anheim
anhin
anlaesslich
ans
anschliessend
anstatt
anstelle
art
auch
auf
auf-das
auffaelligerweise
aufgrund
aufs
aus
ausdruecklich
auseinander
ausgangs
ausgenommen
ausgerechnet
ausschliesslich
aussen
ausser
ausserdem
ausserhalb
ausserordentlich
auswaerts
ausweislich
bald
bar
bedeutend
bedeutet
bedeutsam
beginnen
behufs
bei
bei-der
beide
beidem
beiden
beider
beiderlei
beides
beidseits
beieinander
beim
beinahe
beispiel
beispiele
beispielen
beispielsweise
bekam
bekanntlich
bekennen
bekommen
bekommt
beliebig
bereit
bereits
berichten
besonder
besondere
besonderen
besonderer
besonders
besser
bestehen
betonen
betonte
betraechtlich
betreffend
betreffs
bevor
beziehentlich
bezueglich
bezug
bieten
bietet
bin
binnen
bis
bisher
bisherig
bisherigen
bislang
bisschen
bist
bitte
bleiben
bleibt
blieb
blieben
bloss
brachte
bringen
bzw.
ca.
d.h.
da
dabei
dadurch

dafuer
dagegen
daher
dahinter
damals
damit
danach
daneben
dank
danke
dann
dar
daran
darauf
daraufhin
daraus
darf
darfst
darin
darueber
darum
darunter
das
dasjenige
dass
dasselbe
dasselbige
datum
davon
davor
dazu
dazwischen
de
dein
deine
deinem
deinen
deiner
deines
deinesgleichen
deinige
deinigen
deins
dem
demjenigen
demnaechst
demselben
demselbigen
den
denen
denjenigen
denn
dennoch
denselben
denselbigen
der
derart
derartig
deren
dergleichen
derjenige
derjenigen
derlei
derselbe
derselben
derselbige
derselbigen
derzeit
derzeitig
des
deshalb
desjenigen
desselben
desselbigen
dessen
deswegen
deutlich
dich
die
diejenige
diejenigen
dies
diese
dieselbe
dieselben
dieselbige
dieselbigen
diesem
diesen
dieser
dieses
diesmal
diesseits
ding
dinge
dir
direkt
doch
dort
dran
drauf
drei
dreierlei
dreimal
dritt
dritte
drittel
dritten
drittens
drueber
drum
drunter
du
duerfe
duerfen
duerfest
duerfet
duerft
duerfte
duerften
duerftest
duerftet
durch
durchaus
durchs
durchwegs
durfte
durften
durftest
durftet
eben
ebendies
ebendiese
ebendiesem
ebendiesen
ebendieser
ebendieses
ebenfalls
ebenjene
ebenjenem
ebenjenen
ebenjener
ebenjenes
ebenosweinger
ebenso
ebensoviel
ebensoviele
ebensovielem
ebensovielen
ebensovieler
ebensovieles
ebensowenig
ebensowenigem
ebensowenigen
ebensoweniges
ehemalig
ehemalige
ehemaligen
eher
ehrenhalber
eigen
eigene
eigenen
eigener
eigenes
eigens
eigentlich
eigentliche
eigentlichen
ein
einbegriffen
eine
einem
einen
einer
einerlei
einerseits
eines
einfach
eingangs
eingehend
einig
einige
einigem
einigen
einiger
einigermassen
einiges
einmal
eins
einschliesslich
eis
endlich
entgegen
entlang
entscheiden
entsprechen
entsprechend
entweder
er
erfreulicherweise
erhalten
erklaeren
erklaerte
erst
erste
ersten
erstenmal
erstens
erster
erstere
ersterem
ersteren
ersterer
ersteres
erstes
erstmalig
erstmals
es
esteres
etliche
etlichem
etlichen
etlicher
etliches
etwa
etwas
etwelche
etwelchem
etwelchen
etwelcher
etwelches
euch
euer
euere
euerem
eueren
euerer
eueres
euerm
euern
euers
eure
eurem
euren

eurer
eures
euresgleichen
eurige
eurigen
falls
fast
fern
fernab
ferner
fest
finden
floeten
folgend
folgende
folgenden
folgendermassen
folgt
folgte
fordern
fort
fragen
fragt
frei
freilich
frueh
fruehen
frueher
fruehere
frueheren
fruehestens
fruehzeitig
fuehren
fuenf
fuer
fuer-das
fuers
gab
gaengig
gaengigen
galt
ganz
ganztags
gar
gebe
geben
geblieben
gedurft
gegangen
gegeben
gegebenenfalls
gegen
gegeneinander
gegenueber
gegenwaertig
gehabt
gehalten
gehe
gehen
gehoeren
geht
gekommen
gekonnt
gelegentlich
gelegt
gelingen
gelingt
gelungen
gemacht
gemaess
gemeinsam
gemeinsame
gemeinsamen
gemocht
gemusst
genannt
genannte
genannten
genau
genauer
genauso
generell
genug
gerade
geradezu
gering
gern
gerne
gesagt
gesamt
gesamte
gesamten
gesamthaft
gesehen
gesollt
gewesen
gewiss
gewisse
gewissen
gewollt
geworden
gibt
gilt
ging
gingen
glauben
gleich
gleichermassen
gleichzeitig
groesse
groesser
groessere
groesseren
groesste
groessten
gross
grosse
grossen
grosser
grosses
gruenden
grundsaetzlich
gut
hab
habe
haben
habest
habet
habt
haette
haetten
haettest
haettet
haeufig
haeufigsten
halber
halt
handeln
hast
hat
hatte
hatten
hattest
hattet
hauptsaechlich
haus
heilig
heim
heisst
her
heraus
herein
herum
hervor
heute
hier
hieran
hierauf
hieraus
hierbei
hierdurch
hierein
hierfuer
hiermit
hierueber
hierum
hierunter
hiervon
hiervor
hierzu
hin
hingegen
hinsichtlich
hinter
hinterm
hintern
hinters
hoechstens
hoeher
hoehere
hoeheren
hohe
hohen
hoher
ich
ihm
ihn
ihnen
ihr
ihre
ihrem
ihren
ihrer
ihres
ihresgleichen
ihrige
ihrigem
ihrigen
ihrs
im
immer
immerhin
imstande
in
in-das
in-der
indem
indes
indessen
ineinander
infolge
inklusive
inmitten
inne
innen
inner
inneren
innerhalb
ins
insbesondere
insgesamt
inskuenftig
insofern
international
inzwischen
irgendein
irgendeine
irgendeinem
irgendeinen
irgendeiner
irgendeines
irgendeins
irgendetwas
irgendjemand
irgendwas
irgendwelche
irgendwelchen
irgendwelcher
irgendwem
irgendwen
irgendwer
irgendwessen
irgendwo
ist
ja

jaehrlich
je
jede
jedem
jeden
jedenfalls
jeder
jedermann
jedermanns
jederzeit
jedes
jedesmal
jedoch
jedwede
jedwedem
jedweden
jedweder
jedwedes
jegliche
jeglichem
jeglichen
jeglicher
jegliches
jemand
jenachdem
jene
jenem
jenen
jener
jenes
jenseits
jetzt
jeweilig
jeweiligen
jeweils
kann
kannst
kaum
kehrt
kein
keine
keinem
keinen
keiner
keinerlei
keines
keineswegs
keins
kleine
kleinen
kleiner
kleinere
kleineren
kleines
knapp
koenne
koennen
koennest
koennet
koennt
koennte
koennten
koenntest
koenntet
kommen
kommend
kommenden
kommt
konnte
konnten
konntest
konntet
kuerzlich
kurz
kurze
kurzem
kurzen
laenger
laengs
laengsseits
laengst
laengstens
laesst
laeuft
lag
langem
langen
lass
lassen
laufend
laut
lauter
lediglich
leicht
leichter
leider
letzt
letzte
letzten
letztendlich
letzter
letztere
letzterem
letzteren
letzterer
letzteres
letztlich
lieber
liegen
liegt
liess
liessen
links
live
los
machen
macht
machte
machten
mag
magst
mal
man
manch
manche
manchem
manchen
mancher
mancherlei
manches
manchmal
mangels
mass
massgeblich
maximal
mehr
mehrere
mehreren
mehrerer
mehrerlei
mehrmals
mein
meine
meinem
meinen
meiner
meines
meinesgleichen
meinige
meinigem
meinigen
meins
meint
meinte
meist
meiste
meistem
meisten
meistens
merken
merklich
merkt
merkwuerdig
mich
mindestens
minus
mir
mit
miteinander
mithilfe
mitsamt
mitteilen
mittels
mitten
mittler
mittlerweile
mochte
mochten
mochtest
mochtet
moechte
moechten
moechtest
moechtet
moege
moegen
moegend
moegendst
moegendste
moegendsten
moegest
moeget
moeglich
moeglicherweise
moeglichst
moegt
monatlich
morgen
muesse
muessen
muessend
muessest
muesset
muesst
muesste
muessten
muesstest
muesstet
muss
musst
musste
mussten
musstest
musstet
nach
nachdem
nachhaltig
nachher
naechst
naechste
naechsten
naeher
naemlich
nah
nahe
naheliegenderweise
nahezu
nahm
namens
namhaft
natuerlich
natuerlicherweise
neben
nebeneinander
nebst
nehmen
nein
nennen
neun
nicht
nichts
nie
niemals
niemand
nimmt

nix
noch
nochmals
nun
nunmehr
nur
ob
oben
ober
oberhalb
obgleich
obwohl
obzwar
oder
offenbar
offensichtlich
oft
ohne
ohnehin
optimal
paar
passieren
passiert
per
plus
preis
prioritaer
pro
punkto
quasi
rasch
rasche
recht
rechts
rechtzeitig
reich
reichen
reichlich
reichlichem
reichlicher
reichliches
reicht
rein
relativ
religioes
ruecksichtlich
rund
saemtlich
saemtliche
saemtlichem
saemtlichen
saemtlicher
saemtliches
sagen
sagt
sagte
sah
samt
schade
schaffen
scheinbar
scheinen
scheint
schien
schier
schliesslich
schon
schreiben
schrittweise
schwer
sechs
sechsmal
sehen
sehr
sei
seid
seien
seiest
seiet
sein
seine
seinem
seinen
seiner
seines
seinesgleichen
seinige
seinigen
seins
seit
seitdem
seitens
seither
seitlich
selb
selben
selber
selbst
selten
setzen
setzt
setzte
sich
sicher
sie
sieben
siebenmal
siebziger
siehe
sieht
sind
so
sobald
sodann
sofern
sofort
sog.
sogar
sogenannt
sogenannte
sogenannten
sogleich
solang
solch
solche
solchem
solchen
solcher
solcherlei
solches
soll
solle
sollen
sollend
sollest
sollet
sollst
sollt
sollte
sollten
solltest
solltet
somit
sonder
sondern
sonst
sonstjemand
sonstwas
sonstwem
sonstwen
sonstwer
sooft
sosehr
soviel
soviele
sovielem
sovielen
sovieler
sovieles
sowas
soweit
sowie
sowohl
spaet
spaeter
spaeteren
spaetestens
sprechen
staendig
stand
standen
stark
statt
stattdessen
stecken
steckt
stehen
stehend
steht
stellen
stellt
stellte
stets
teil
teilen
teilte
teilweise
terminlich
toll
trat
trotz
trotzdem
tun
tut
typisch
ueber
ueberall
ueberaus
ueberdies
ueberhand
ueberhaupt
ueberm
uebermorgen
uebern
uebers
ueblich
ueblicherweise
uebrig
uebrigen
uebrigens
um
umfangreich
umfangreiche
umfangreichen
umfangreiches
umgekehrt
ums
unangesehen
unbedingt
unbeschadet
und
unerachtet
unfern
ungeachtet
ungefaehr
unlaengst
unrem
uns
unser
unsere
unsereinem
unsereinen
unsereiner
unsereines
unsereins
unserem
unseren
unserer
unseres
unseresgleichen
unserige
unserigen
unserm
unsern
unsers

unsre
unsrem
unsren
unsrer
unsres
unsrige
unsrigen
unten
unter
untereinander
unterhalb
unterm
untern
unters
unterwegs
unweigerlich
unweit
unwesentlich
usw.
van
vergehen
vermag
vermittels
vermittelst
vermoege
vermutlich
verschieden
verschiedentlich
versuchsweise
versus
verwenden
verwendet
verwendete
verwendeten
verwendung
via
viel
viele
vielem
vielen
vieler
vielerlei
vieles
vielfach
vielleicht
vielmals
vielmehr
vielzahl
vier
viermal
viert
viertens
voellig
voll
voller
vollstaendig
vom
von
von-der
vor
vorbehaltlich
vorbei
vorerst
vorher
vorlieb
vorliegend
vorliegende
vorliegenden
vorm
vormals
vorne
vornehmlich
vors
vorsitzen
waehren
waehrend
waere
waeren
waerest
waeret
waerst
waert
wahrlich
wann
war
waren
warest
warst
wart
warum
was
weder
weg
wegen
weh
weil
weis
weise
weisen
weiss
weist
weit
weitaus
weiter
weitere
weiteren
weiterer
weiteres
weiterhin
welch
welche
welchem
welchen
welcher
welches
wem
wen
wenden
wendet
wenig
wenige
wenigem
wenigen
weniger
weniges
wenigste
wenigstem
wenigsten
wenigstens
wenigster
wenigstes
wenn
wer
werde
werden
wert
wesentlich
weshalb
wessen
wessentwegen
wessentwillen
weswegen
wett
wichtig
wider
wie
wieder
wiederum
wieso
wieviel
wievielerlei
wievielmal
wieweit
will
willen
willst
wir
wird
wirklich
wo
wobei
wodurch
woechentlich
wofuer
wogegen
woher
woherum
wohin
wohinauf
wohinaus
wohinein
wohinter
wohinunter
wohl
wolle
wollen
wollest
wollet
wollt
wollte
wollten
wolltest
wolltet
womit
wonach
woneben
worab
woran
worauf
woraufhin
woraus
worden
worein
worin
worueber
worum
worunter
wovon
wovor
wozu
wozwischen
wuerde
wuerden
wunder
wurde
wurden
z.B.
zahlreich
zahlreiche
zahlreichen
zahlreicher
zehn
zeigen
zeigt
zeigte
zeit
zeitlich
zentral
ziehen
zieht
ziemlich
zog
zu
zu-der
zu-die
zudem
zueinander
zuerst
zufaelligerweise
zufolge
zugaenglich
zugleich
zugunsten
zugute
zukuenftig
zuletzt
zuliebe
zum
zumal
zumindest
zunaechst
zunichte
zur
zurzeit
zusaetzlich

zusaetzliche
zusammen
zustande
zuungunsten
zuviel
zuviele
zuvielem
zuvielen
zuvieler
zuvieles
zuvor
zuwenig
zuwenige
zuwenigem
zuwenigen
zuweniger
zuweniges
zuwider
zuzueglich
zwanzig
zwar
zwecks
zwei
zweierlei
zweifellos
zweimal
zweit
zweite
zweiten
zweitens
zwischen
zwoelf

C.2 English

a
ability
able
aboard
about
above
absolute
absolutely
across
act
acts
actual
actually
add
additional
additionally
after
afterwards
again
against
ago
ahead
aimless
aimlessly
ain't
al
albeit
align
all
allow
almost
along
alongside
already
also
alternate
alternately
although
always
am
amid
amidst
among
amongst
&
an
and
announce
announced
announcement
announces
another
anti
any
anyone
anything
appaling
appalingly
appear
appeared
appears
are
aren't
around
as
ask
asked
asking
asks
at
await
awaited
awaits
awaken
awakened
awakens
aware
away
b
back
backed
backing
backs
be
became
because
become
becomes
becoming
been
before
began
begin
begins
behind
being
believe
believed
between
bgcolor
border
both
brang
bring
brings
brought
build
builds
built
busy
but
by
c
call
called
calling
calls
can
cannot
can't
carried
carries
carry
carrying
cellspacing
center
certainly
change
changed
changes
choose
chooses
chose
clearly
close
closed
closes
closing
com
come
comes
coming
consider
considerable
considering
contains
could
couldn
couldn't
d
dare
daren
day
days
despite
did
didn
didn't
do
does
doesn
doesn't
doing
done
don't
down
downward
downwards
e
each
eight
either
else
elsewhere
especially
etc
even
eventually
ever
every
everybody
everyone
exactly
example
examples
f
far
feel
felt
few

ffc0c
final
finally
find
first
five
float
for
found
four
fourth
from
gave
get
gets
getting
give
gives
go
goes
going
gone
good
got
great
h
had
happen
happened
happens
has
have
haven't
he
he'd
held
her
here
hereby
heretofore
herewith
hers
herself
he's
high
him
himself
his
hitherto
homepage
hour
hours
how
however
i
i'd
if
ii
iii
i'll
i'm
important
in
inc
include
included
includes
including
inside
into
is
isn
isn't
it
it'd
it'll
it's
its
itself
iv
i've
j
just
k
kind
kinds
l
la
larger
largest
last
later
latest
le
least
leave
leaves
leaving
les
less
let
like
ll
lya
m
made
main
mainly
make
makes
making
man
many
margin-left
may
me
means
meant
meanwhile
men
methods
might
missed
more
moreover
most
mostly
move
moved
moving
mr
mrs
much
must
mustn
my
myself
need
needs
neither
never
new
newer
news
night
nights
nine
no
non
none
nor
not
now
o
of
off
often
old
on
once
one
only
or
other
our
out
over
own
owns
p
particularly
per
percent
present
presentation
presented
presenter
presenting
presents
primarily
put
q
quickly
r
ran
rather
recent
remain
remaining
respond
responded
responding
responds
return
right
run
running
runs
s
said
same
say
says
see
seek
seeking
seeks
seen
send
sent
set
sets
seven
several
she
she's
should
shouldn
shown
side
since
six
sixes
slow
slowed
slows
small
smaller
so
some
someone
something
somewhat
somewhere
soon
sought
spread
stay
stayed
still
substantially
such
suppose
t
take

taken
takes
taking
tell
tells
th
than
that
the
their
them
themselves
then
there
thereby
therefore
these
they
they'd
they're
they've
thi
thing
things
this
those
though
three
through
throughout
thus
to
to-day
together
too
took
toward
towards
tried
tries
try
trying
two
u
unable
under
underneath
undid
undo
undoes
undone
undoubtedly
undue
unfortunately
unless
unnecessarily
unofficially
unsure
until
unusually
up
upon
upward
us
use
used
uses
using
usual
usually
v
various
ve
very
via
view
viewed
w
wait
waited
waits
want
wanted
wants
was
wasn
wasn't
watched
watching
way
ways
we
weblink
we'd
we'll
well
went
we're
were
we've
what
whatever
when
whenever
where
whereever
whether
which
whichever
while
who
whoever
whom
whomsoever
whose
whosever
who've
why
wide
wider
will
with
without
won
won't
would
wouldn
wouldn't
wow
wows
www
x
xii
xiii
xiv
xix
xv
xvi
xvii
xviii
xx
y
yeah
year
-year-old
you
you'll
your
you're
your's
yours
yourself
yourselves

Appendix D

Experimental Result Tables

This appendix collects experimental result tables omitted from the main body of the text. All of them have at least partially been illustrated by figures in the text.

D.1 Experiments in Chapter 4 (Setup)

rowmodel	colmodel	SPRINGER	AMAZON	SDA	WIKI	NZZ
NONE	NONE	0.789[0.008]	0.644[0.002]	0.636[0.000]	0.588[0.012]	0.581[0.017]
LOG	NONE	0.829[0.041]	0.649[0.001]	0.627[0.001]	0.583[0.011]	0.480[0.000]
MAXTF	NONE	0.804[0.026]	0.622[0.003]	0.581[0.000]	0.558[0.008]	0.476[0.000]
SQRT	NONE	0.790[0.015]	0.619[0.002]	0.580[0.001]	0.557[0.011]	0.475[0.000]
NONE	IDF	0.491[0.008]	0.496[0.002]	0.522[0.001]	0.410[0.008]	0.402[0.022]
LOG	IDF	0.491[0.005]	0.497[0.003]	0.501[0.001]	0.408[0.004]	0.370[0.016]
MAXTF	IDF	0.486[0.004]	0.491[0.005]	0.490[0.000]	0.414[0.012]	0.371[0.001]
SQRT	IDF	0.487[0.004]	0.486[0.005]	0.491[0.001]	0.407[0.006]	0.369[0.000]

Table D.1: **Evaluation of Cluto weighting models**, with the *rbr(largess)* algorithm. (The table shows entropy values and in square brackets standard deviations.) [This table belongs to Section 4.4.2]

rowmodel	colmodel	SPRINGER	AMAZON	SDA	WIKI	NZZ
NONE	NONE	0.488[0.002]	0.486[0.005]	0.495[0.000]	0.400[0.006]	0.364[0.000]
LOG	NONE	0.582[0.012]	0.550[0.004]	0.517[0.000]	0.457[0.008]	0.441[0.028]
MAXTF	NONE	0.653[0.007]	0.609[0.004]	0.535[0.001]	0.539[0.009]	0.417[0.017]
SQRT	NONE	0.617[0.004]	0.583[0.005]	0.526[0.000]	0.491[0.007]	0.456[0.018]
NONE	IDF	0.422[0.018]	0.463[0.005]	0.421[0.001]	0.408[0.019]	0.349[0.005]
LOG	IDF	0.463[0.003]	0.483[0.004]	0.470[0.000]	0.394[0.007]	0.361[0.001]
MAXTF	IDF	0.474[0.005]	0.482[0.005]	0.478[0.000]	0.400[0.016]	0.376[0.023]
SQRT	IDF	0.468[0.004]	0.481[0.003]	0.473[0.000]	0.392[0.009]	0.391[0.034]

Table D.2: **Double-weighting**. Evaluation of CLUTO's built-in weighting models *after* prior application of an external LOG-IDF weighting scheme to the data. (The table shows entropy values and in square brackets standard deviations.) [Section 4.4.2.1]

	external		CLUTO						
	local	global	local	global	SPRINGER	AMAZON	SDA	WIKI	NZZ
1	—	—	—	—	0.789[0.008]	0.644[0.002]	0.636[0.000]	0.588[0.012]	0.581[0.017]
2	—	—	—	IDF_2	0.491[0.008]	0.496[0.002]	0.522[0.001]	0.410[0.008]	0.402[0.022]
3	—	IDF_2	—	—	0.490[0.005]	0.497[0.002]	0.522[0.001]	0.408[0.007]	0.402[0.021]
4	—	IDF_n	—	—	0.491[0.009]	0.497[0.004]	0.522[0.000]	0.407[0.005]	0.397[0.019]
5	—	—	LOG_2	IDF_2	0.491[0.005]	0.497[0.003]	0.501[0.001]	0.408[0.004]	0.370[0.016]
6	LOG_2	IDF_2	—	—	0.489[0.003]	0.497[0.002]	0.502[0.001]	0.408[0.022]	0.381[0.025]
7	LOG_n	IDF_n	—	—	0.488[0.002]	0.486[0.005]	0.495[0.000]	0.400[0.006]	0.364[0.000]
8	—	$(\text{IDF}_2)^2$	—	—	0.416[0.020]	0.475[0.004]	0.428[0.003]	0.417[0.025]	0.438[0.020]
9	—	$(\text{IDF}_n)^2$	—	—	0.416[0.018]	0.474[0.008]	0.430[0.003]	0.403[0.019]	0.444[0.007]
10	LOG_n	IDF_n	—	IDF_2	0.422[0.018]	0.463[0.005]	0.421[0.001]	0.408[0.019]	0.349[0.005]
11	LOG_n	IDF_nIDF_2	—	—	0.424[0.018]	0.463[0.004]	0.422[0.001]	0.413[0.017]	0.350[0.005]
12	LOG_2	IDF_2	—	IDF_2	0.417[0.019]	0.467[0.007]	0.422[0.004]	0.394[0.017]	0.362[0.006]
13	LOG_2	$(\text{IDF}_2)^2$	—	—	0.414[0.019]	0.465[0.007]	0.421[0.004]	0.407[0.023]	0.360[0.009]
14	LOG_n	$(\text{IDF}_n)^2$	—	—	0.414[0.019]	0.463[0.004]	0.421[0.004]	0.412[0.010]	0.351[0.005]

Table D.3: **Evaluation of different IDF and IDF^2 variants.** IDF_n and LOG_n refer to formulae using the natural logarithm, IDF_2 and LOG_2 to those with the binary logarithm. The differences between rows 2/3, 5/6, 10/11 and 12/13 can be explained by rounding errors in the intermediate matrix emerging from the external weighting step. In principal the members of each of these pairs are exactly equivalent. [Section 4.4.2.2]

		k1b	**sports**	**tr23**
	# of documents	2,340	8,580	204
	# labels	6	7	6
	# types in corpus	21,839	126,373	5,832
	# tokens in corpus	552,213	1,869,981	493,387
	avg. text length	236	218	2419[a]

	external		CLUTO				
	local	global	local	global			
1	—	—	—	—	0.248[0.000]	0.353[0.006]	0.494[0.000]
2	—	—	—	IDF_2	0.192[0.017]	0.198[0.007]	0.411[0.005]
3	—	IDF_2	—	—	0.194[0.017]	0.195[0.005]	0.411[0.005]
4	—	IDF_n	—	—	0.187[0.012]	0.194[0.005]	0.411[0.005]
5	—	—	LOG_2	IDF_2	0.190[0.016]	0.214[0.022]	0.458[0.000]
6	LOG_2	IDF_2	—	—	0.190[0.016]	0.209[0.024]	0.458[0.000]
7	LOG_n	IDF_n	—	—	0.211[0.007]	0.211[0.024]	0.464[0.000]
8	—	$(\text{IDF}_2)^2$	—	—	0.170[0.014]	0.221[0.020]	0.411[0.018]
9	—	$(\text{IDF}_n)^2$	—	—	0.171[0.014]	0.219[0.015]	0.404[0.017]
10	LOG_n	IDF_n	—	IDF_2	0.192[0.023]	0.265[0.031]	0.431[0.020]
11	LOG_n	IDF_nIDF_2	—	—	0.191[0.027]	0.253[0.024]	0.432[0.016]
12	LOG_2	IDF_2	—	IDF_2	0.178[0.016]	0.237[0.002]	0.435[0.023]
13	LOG_2	$(\text{IDF}_2)^2$	—	—	0.183[0.021]	0.238[0.001]	0.435[0.023]
14	LOG_n	$(\text{IDF}_n)^2$	—	—	0.191[0.021]	0.246[0.000]	0.429[0.015]

[a]The average text length for tr23 is strongly influenced by a small number of very large documents. Barring the twelve longest documents, the average is reduced to 677 tokens. Barring another twelve documents it is even reduced to 341.

Table D.4: **Experiments with three standard sets.** Different weighting schemes applied to the three test collections *k1b*, *sports* and *tr23* coming as part of the CLUTO standard distribution. Scores in the upper half refer to normal weighting, scores in the lower half to IDF^2 weightings. [Section 4.4.2.2]

D.2 Experiments in Chapter 5 (Matrix Reduction)

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOW	0.476[0.016]	0.460[0.004]	0.474[0.003]	0.392[0.012]	0.348[0.001]
BOW_{stop}	0.449[0.015]	0.466[0.006]	0.436[0.002]	0.416[0.020]	0.346[0.001]
BOW_{stem}	0.408[0.004]	0.458[0.009]	0.463[0.002]	0.386[0.010]	0.346[0.002]
$BOW_{stop,\ stem}$	0.410[0.010]	0.468[0.006]	0.431[0.002]	0.416[0.019]	0.348[0.002]

Table D.5: **Baseline and variants.** BOW = standard bag-of-words; BOW_{stop} = bag-of-words after stopword removal; $BOW_{stop,\ stem}$ = bag-of-words after stopword removal and stemming. [Section 5.1.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOW	0.476[0.016]	0.460[0.004]	0.474[0.003]	0.392[0.012]	0.348[0.001]
BOL	0.421[0.015]	0.448[0.007]	0.456[0.001]	0.388[0.007]	0.349[0.005]
BOW_{stem}	0.408[0.004]	0.458[0.009]	0.463[0.002]	0.386[0.010]	0.346[0.002]
BOL_{stem}	0.407[0.004]	0.454[0.010]	0.455[0.001]	0.390[0.007]	0.357[0.004]

Table D.6: **Bag-of-lemmata versus bag-of-words.** [Section 5.2.2]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
100%	0.421	0.448	0.456	0.388	0.349
prune to 99%	0.421	0.483	0.428	0.409	0.346
prune to 95%	0.443	0.498	0.476	0.416	0.366
prune to 90%	0.488	0.518	0.497	0.433{1}	0.400
prune to 85%	0.530	0.523	0.506	0.454{1}	0.404

Table D.7: **Global pruning with similarity preservation,** using CLUTO's *–colprune* parameter. (In curly brackets: number of documents that could not be clustered.) [Section 5.3.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
All	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
Shared	0.420 (90%)	0.450 (97%)	0.457 (98%)	0.389 (94%)	0.349 (97%)
0 – 10 %	0.416 (69%)	0.460 (72%)	0.448 (67%)	0.394 (76%)	0.353 (64%)
0.005 – 10 %		0.476 (66%)	0.442 (63%)	0.390 (68%)	0.355 (62%)
0.01 – 10 %		0.482 (64%)	0.438 (61%)	0.404 (64%)	0.347 (60%)
0.05 – 10 %	0.412 (60%)	0.495 (56%)	0.434 (55%)	0.414 (55%)	0.351 (54%)
0.1 – 10 %	0.436 (54%)	0.498 (51%)	0.464 (51%)	0.421 (50%)	0.356 (51%)
0 – 5 %	0.419 (58%)	0.469 (63%)	0.418 (58%)	0.423 (67%)	0.356 (53%)
0.005 – 5 %		0.473 (57%)	0.416 (54%)	0.428 (59%)	0.356 (50%)
0.01 – 5 %		0.480 (55%)	0.407 (52%)	0.416 (55%)	0.351 (48%)
0.05 – 5 %	0.430 (48%)	0.491 (47%)	0.403 (46%)	0.414 (46%)	0.353 (43%)
0.1 – 5 %	0.432 (42%)	0.498 (42%)	0.410 (42%)	0.416'(42%)	0.360 (39%)
0 – 1 %	0.453 (37%)	0.484 (43%)	0.392 (36%)	0.427 (47%)	0.357 (31%)
0.005 – 1 %		0.489 (37%)	0.381 (32%)	0.431 (39%)	0.353 (29%)
0.01 – 1 %		0.493 (35%)	0.383 (31%)	0.435'(35%)	0.351 (27%)
0.05 – 1 %	0.451 (27%)	0.510'(27%)	0.397'(24%)	0.439'(26%)	0.362 (21%)
0.1 – 1 %	0.473'(21%)	0.525'(22%)	0.416'(21%)	0.447'(21%)	0.377 (18%)
0 – 0.5 %	0.497'(29%)	0.513 (35%)	0.407 (28%)	0.434'(40%)	0.356 (25%)
0.005 – 0.5 %		0.513'(29%)	0.415'(24%)	0.432'(31%)	0.356 (22%)
0.01 – 0.5 %		0.514'(27%)	0.410'(23%)	0.432'(28%)	0.350 (20%)
0.05 – 0.5 %	0.504'(20%)	0.536'(19%)	0.422'(16%)	0.455'(19%)	0.376 (15%)
0.1 – 0.5 %	0.543'(14%)	0.568'(14%)	0.446'(13%)	0.485'(14%)	0.388 (11%)

Table D.8: **Global pruning with upper and lower bounds** on document frequency. Percentage numbers in parentheses indicate how many of the non-zero elements were still left after pruning. Apostrophes indicate that some documents could not be clustered because all features had been eliminated. The number of documents thus lost was far less than 1% in all cases. SPRINGER results for lower bounds 0.005% and 0.01% have been omitted—owing to the comparatively small number of documents, these boundaries are equivalent to no boundary at all. [Section 5.3.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
All features (shared and unique)					
$\alpha \to \infty$	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
$\alpha = 300$	0.424 (100%)	0.456 (97%)	0.457 (100%)	0.389 (85%)	0.386 (78%)
$\alpha = 250$	0.424 (100%)	0.457 (95%)	0.458 (100%)	0.393 (81%)	0.400 (67%)
$\alpha = 200$	0.423 (100%)	0.461 (91%)	0.453 (97%)	0.390 (75%)	0.418 (54%)
$\alpha = 150$	0.426 (100%)	0.454 (82%)	0.451 (90%)	0.388 (66%)	0.429 (41%)
$\alpha = 100$	0.422 (99%)	0.468 (69%)	0.446 (74%)	0.392 (53%)	0.450 (27%)
$\alpha = 50$	0.429 (74%)	0.485 (42%)	0.429 (42%)	0.412'(31%)	0.481 (14%)
$\alpha = 25$	0.457'(38%)	0.509'(22%)	0.446 (21%)	0.454'(17%)	0.630'(7%)
$\alpha = 10$	0.851'(15%)	0.660'(9%)	0.606'(8%)	0.646'(7%)	0.721'(3%)
Shared features only					
$\alpha \to \infty$	0.420 (90%)	0.450 (97%)	0.457 (98%)	0.389 (94%)	0.349 (97%)
$\alpha = 300$	0.423 (90%)[a]	0.453 (94%)	0.456 (98%)	0.397 (82%)	0.374 (77%)
$\alpha = 250$	0.423 (90%)[a]	0.456 (92%)	0.457 (98%)	0.390 (78%)	0.395 (67%)
$\alpha = 200$	0.423 (90%)[a]	0.459 (88%)	0.454 (95%)	0.386 (73%)	0.412 (54%)
$\alpha = 150$	0.423 (90%)[a]	0.451 (80%)	0.451 (88%)	0.393 (64%)	0.429 (41%)
$\alpha = 100$	0.420 (90%)	0.470 (67%)	0.447 (73%)	0.382 (52%)	0.446 (27%)
$\alpha = 50$	0.424 (73%)	0.484 (42%)	0.428 (42%)	0.407 (31%)	0.451 (14%)
$\alpha = 25$	0.414 (38%)	0.503 (22%)	0.437 (21%)	0.438 (17%)	0.598 (7%)
$\alpha = 10$	0.575 (15%)	0.614 (9%)	0.540 (8%)	0.553 (7%)	0.676 (3%)

[a]The SPRINGER results for $\alpha \geq 150$ are slightly different from that for $\alpha \to \infty$ even though essentially the same matrix was used in all instances. The difference is explained by the variability of the non-deterministic clustering results resulting from a different arrangement of the matrix columns.

Table D.9: **Local pruning:** keeping only the α most frequent features of each document (after LOG-IDF weighting). The first half of the table shows the procedure with all features. In the second half those features not occurring in other documents ("unique features") were removed before making the selection. The differences are not big except for the cases where documents would have been lost altogether. (In parentheses: percentage of non-zero elements left after reduction.) [Section 5.3.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
LSA-5	0.633[0.000]	n/a	n/a	n/a	n/a
LSA-25	0.541[0.004]	n/a	n/a	n/a	n/a
LSA-50	0.542[0.003]	n/a	n/a	n/a	n/a
LSA-75	0.536[0.005]	n/a	n/a	n/a	n/a
LSA-100	0.527[0.006]	n/a	n/a	n/a	n/a
LSA-125	0.538[0.006]	n/a	n/a	n/a	n/a
LSA-150	0.541[0.008]	n/a	n/a	n/a	n/a
LSA-175	0.544[0.006]	n/a	n/a	n/a	n/a
LSA-200	0.535[0.006]	n/a	n/a	n/a	n/a
LSA-225	0.546[0.016]	n/a	n/a	n/a	n/a
LSA-250	0.547[0.023]	n/a	n/a	n/a	n/a
LSA-275	0.546[0.024]	n/a	n/a	n/a	n/a
LSA-300	0.557[0.029]	n/a	n/a	n/a	n/a

Table D.10: **Latent Semantic Analysis.** LSA-ρ refers to the clustering experiment with the ρ SVD-dimensions that had the highest corresponding eigenvalues. (For LSA experiments CLUTO's *–cstype* parameter was switched from *largeSS* to *large* as the former is less suitable for dense matrices such as those arising from LSA.) [Section 5.3.2]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOW	0.476 (100%)	0.460 (100%)	0.474 (100%)	0.392 (100%)	0.348 (100%)
BOW_{stop}	0.449 (59%)	0.466 (58%)	0.436 (57%)	0.416 (65%)	0.346 (64%)
$\text{BOW}_{\text{stop, stem}}$	0.410 (58%)	0.468 (55%)	0.431 (54%)	0.416 (61%)	0.348 (60%)
$\text{BOW}_{\text{stop[Google]}}$	0.461 (72%)	0.462 (76%)	0.467 (75%)	0.398 (81%)	0.348 (84%)
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
BOL_{stop}	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)
$\text{BOL}_{\text{stop, stem}}$	0.419 (60%)	0.468 (59%)	0.423 (59%)	0.411 (66%)	0.348 (66%)
$\text{BOL}_{\text{stop[Google]}}$	0.437 (76%)	0.459 (80%)	0.451 (79%)	0.390 (84%)	0.347 (88%)

Table D.11: **Stopword removal** with manually compiled stoplist. (In parentheses: percentage of non-zero elements left after reduction.) [Section 5.4.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
df	0.519 (40%)	0.485 (50%)	0.419 (46%)	0.418 (57%)	0.357 (58%)
discr	0.393 (48%)	0.458 (61%)	0.427 (65%)	0.389 (66%)	0.352 (70%)
χ^2	0.403 (99%)	0.451 (99%)	0.449 (94%)	0.388 (99%)	0.351 (90%)
E	0.392 (52%)	0.459 (83%)	0.432 (79%)	0.398 (71%)	0.352 (81%)
E'	0.419 (86%)	0.451 (99%)	0.456 (99%)	0.388 (99%)	0.348 (96%)
E''	0.388 (47%)	0.457 (62%)	0.401 (57%)	0.393 (67%)	0.355 (65%)
KL	0.396 (46%)	0.462 (60%)	0.423 (63%)	0.389 (65%)	0.352 (68%)
WKL	0.458 (41%)	0.473 (52%)	0.401 (49%)	0.397 (59%)	0.356 (60%)

Table D.12: **"Self-validation" of different stopword discrimination measures (averages).** The numbers show the averages for the nine different α values. See Table D.13 for the detailed results. [Section 5.4.2.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
df (50)	0.424 (72%)	0.450 (80%)	0.454 (77%)	0.399 (83%)	0.349 (88%)
df (100)	0.405 (64%)	0.460 (72%)	0.448 (69%)	0.387 (78%)	0.348 (81%)
df (250)	0.434 (52%)	0.469 (63%)	0.421 (59%)	0.397 (69%)	0.349 (72%)
df (500)	0.441 (43%)	0.474 (54%)	0.411 (50%)	0.420 (62%)	0.354 (63%)
df (1000)	0.461 (34%)	0.471 (46%)	0.393 (41%)	0.422 (54%)	0.362 (53%)
df (1500)	0.496'(29%)	0.486 (41%)	0.394 (36%)	0.422 (49%)	0.363 (47%)
df (2000)	0.534'(25%)	0.498 (37%)	0.415 (32%)	0.440'(45%)	0.368 (43%)
df (2500)	0.594'(23%)	0.512'(34%)	0.424 (30%)	0.433'(42%)	0.354 (40%)
df (5000)	0.887'(16%)	0.541'(26%)	0.411'(22%)	0.439'(34%)	0.363 (30%)
discr (50)	0.413 (76%)	0.455 (84%)	0.453 (85%)	0.385 (85%)	0.351 (89%)
discr (100)	0.421 (70%)	0.458 (80%)	0.451 (80%)	0.384 (80%)	0.348 (85%)
discr (250)	0.410 (62%)	0.460 (73%)	0.448 (74%)	0.395 (74%)	0.350 (80%)
discr (500)	0.410 (53%)	0.459 (66%)	0.438 (67%)	0.392 (69%)	0.346 (75%)
discr (1000)	0.363 (44%)	0.458 (59%)	0.423 (63%)	0.398 (63%)	0.351 (69%)
discr (1500)	0.372 (38%)	0.455 (54%)	0.418 (58%)	0.393 (60%)	0.354 (65%)
discr (2000)	0.371 (34%)	0.453 (50%)	0.413 (56%)	0.384 (57%)	0.356 (61%)
discr (2500)	0.379 (31%)	0.459 (47%)	0.411 (54%)	0.387 (55%)	0.355 (59%)
discr (5000)	0.397'(23%)	0.464'(37%)	0.391 (47%)	0.386 (48%)	0.354 (51%)
χ^2 (50)	0.417 (91%)	0.452 (100%)	0.456 (99%)	0.385 (100%)	0.351 (96%)
χ^2 (100)	0.414 (86%)	0.452 (100%)	0.457 (99%)	0.390 (100%)	0.352 (95%)
χ^2 (250)	0.401 (73%)	0.446 (100%)	0.455 (97%)	0.386 (100%)	0.353 (94%)
χ^2 (500)	0.381 (45%)	0.449 (100%)	0.456 (97%)	0.385 (100%)	0.349 (92%)
χ^2 (1500)	0.434'(38%)	0.450 (99%)	0.449 (94%)	0.389 (99%)	0.349 (90%)
E (50)	0.415 (97%)	0.453 (99%)	0.457 (100%)	0.391 (98%)	0.349 (99%)
E (100)	0.410 (87%)	0.460 (98%)	0.456 (100%)	0.397 (94%)	0.349 (99%)
E (250)	0.410 (67%)	0.454 (97%)	0.451 (99%)	0.410 (83%)	0.349 (98%)
E (500)	0.388 (55%)	0.459 (94%)	0.447 (95%)	0.405 (74%)	0.353 (95%)
E (1000)	0.369 (42%)	0.466 (90%)	0.435 (81%)	0.406 (67%)	0.353 (81%)
E (1500)	0.351 (36%)	0.461 (83%)	0.425 (69%)	0.403 (61%)	0.354 (73%)
E (2000)	0.389 (31%)	0.464 (73%)	0.415 (62%)	0.398 (58%)	0.355 (69%)
E (2500)	0.390 (29%)	0.453 (67%)	0.413 (59%)	0.387 (55%)	0.349 (65%)
E (5000)	0.407'(21%)	0.463'(47%)	0.386 (47%)	0.382 (47%)	0.354 (52%)
E' (50)	0.420 (95%)	0.452 (100%)	0.456 (99%)	0.391 (100%)	0.350 (97%)
E' (100)	0.413 (94%)	0.449 (100%)	0.457 (99%)	0.384 (100%)	0.352 (97%)
E' (250)	0.428 (93%)	0.448 (100%)	0.457 (99%)	0.390 (100%)	0.349 (97%)
E' (500)	0.425 (91%)	0.451 (100%)	0.455 (99%)	0.389 (100%)	0.348 (97%)
E' (1000)	0.427 (90%)	0.452 (100%)	0.456 (98%)	0.384 (99%)	0.347 (97%)
E' (1500)	0.429 (85%)	0.445 (99%)	0.457 (98%)	0.394 (99%)	0.346 (96%)
E' (2000)	0.415 (81%)	0.451 (99%)	0.457 (98%)	0.388 (99%)	0.349 (96%)
E' (2500)	0.412 (77%)	0.449 (99%)	0.456 (98%)	0.390 (99%)	0.345 (96%)
E' (5000)	0.404 (67%)	0.460 (98%)	0.452 (98%)	0.385 (99%)	0.346 (95%)

continued on next page

continued from previous page

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
E" (50)	0.418 (80%)	0.455 (91%)	0.451 (86%)	0.391 (96%)	0.349 (97%)
E" (100)	0.404 (65%)	0.457 (80%)	0.448 (71%)	0.389 (91%)	0.352 (93%)
E" (250)	0.398 (55%)	0.455 (67%)	0.428 (61%)	0.389 (77%)	0.348 (82%)
E" (500)	0.384 (49%)	0.457 (62%)	0.409 (56%)	0.397 (64%)	0.354 (65%)
E" (1000)	0.366 (42%)	0.456 (57%)	0.401 (53%)	0.401 (59%)	0.356 (56%)
E" (1500)	0.367 (38%)	0.457 (55%)	0.378 (50%)	0.396 (56%)	0.356 (51%)
E" (2000)	0.367 (36%)	0.457 (53%)	0.366 (48%)	0.387 (54%)	0.359 (49%)
E" (2500)	0.382 (32%)	0.457 (51%)	0.364 (46%)	0.406 (53%)	0.358 (47%)
E" (5000)	0.405'(25%)	0.459 (45%)	0.360 (43%)	0.381'(47%)	0.359 (42%)
KL (50)	0.425 (75%)	0.452 (84%)	0.453 (84%)	0.387 (85%)	0.350 (88%)
KL (100)	0.416 (69%)	0.452 (78%)	0.451 (79%)	0.384 (80%)	0.354 (84%)
KL (250)	0.411 (60%)	0.457 (71%)	0.446 (72%)	0.390 (74%)	0.347 (78%)
KL (500)	0.393 (51%)	0.466 (64%)	0.438 (66%)	0.388 (68%)	0.350 (72%)
KL (1000)	0.380 (40%)	0.467 (56%)	0.419 (60%)	0.389 (62%)	0.348 (66%)
KL (1500)	0.385 (35%)	0.466 (52%)	0.415 (57%)	0.394 (58%)	0.349 (62%)
KL (2000)	0.391 (31%)	0.465 (49%)	0.415 (54%)	0.395 (55%)	0.355 (59%)
KL (2500)	0.409 (29%)	0.468 (45%)	0.410 (52%)	0.396 (53%)	0.356 (56%)
KL (5000)	0.351'(21%)	0.465 (37%)	0.359 (42%)	0.377 (45%)	0.355 (46%)
WKL (50)	0.413 (73%)	0.453 (80%)	0.453 (78%)	0.389 (84%)	0.351 (88%)
WKL (100)	0.416 (66%)	0.457 (74%)	0.449 (71%)	0.390 (78%)	0.352 (82%)
WKL (250)	0.404 (55%)	0.460 (64%)	0.431 (62%)	0.391 (70%)	0.347 (73%)
WKL (500)	0.388 (45%)	0.468 (56%)	0.414 (54%)	0.394 (64%)	0.348 (65%)
WKL (1000)	0.406 (36%)	0.466 (48%)	0.383 (45%)	0.394 (56%)	0.354 (56%)
WKL (1500)	0.429 (30%)	0.480 (42%)	0.371 (40%)	0.397 (51%)	0.355 (50%)
WKL (2000)	0.454'(26%)	0.484 (39%)	0.361 (36%)	0.404 (47%)	0.355 (46%)
WKL (2500)	0.466'(24%)	0.484 (36%)	0.367 (33%)	0.411'(44%)	0.363 (42%)
WKL (5000)	0.747'(17%)	0.508'(28%)	0.380'(24%)	0.407'(36%)	0.374 (33%)
BOL_{stop}	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)

Table D.13: **Self-validation of different stopword discrimination measures.** For each data set stopwords were identified according to the different measures and then applied to the same data set. [Section 5.4.2.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
df (50)	0.420 (72%)	0.446 (77%)	0.450 (76%)	0.381 (82%)	0.351 (85%)
df (100)	0.415 (67%)	0.456 (70%)	0.445 (68%)	0.390 (76%)	0.352 (78%)
df (250)	0.398 (58%)	0.453 (59%)	0.432 (58%)	0.391 (67%)	0.352 (67%)
df (500)	0.424 (48%)	0.459 (52%)	0.418 (50%)	0.408 (59%)	0.355 (57%)
df (1000)	0.409 (39%)	0.466 (44%)	0.416 (40%)	0.415 (51%)	0.349 (47%)
df (1500)	0.406 (34%)	0.482'(39%)	0.413 (35%)	0.433 (47%)	0.347 (41%)
df (2000)	0.432 (30%)	0.489'(36%)	0.423 (32%)	0.435'(43%)	0.364 (38%)
df (2500)	0.437 (28%)	0.503'(33%)	0.420 (29%)	0.435'(41%)	0.368 (35%)
df (5000)	0.490'(22%)	0.559'(26%)	0.437'(22%)	0.451'(34%)	0.400 (27%)
discr (50)	0.419 (70%)	0.453 (76%)	0.448 (75%)	0.387 (81%)	0.350 (85%)
discr (100)	0.419 (64%)	0.460 (71%)	0.445 (68%)	0.394 (76%)	0.351 (79%)
discr (250)	0.426 (56%)	0.461 (64%)	0.424 (61%)	0.398 (68%)	0.353 (69%)
discr (500)	0.408 (50%)	0.457 (57%)	0.415 (54%)	0.403 (62%)	0.348 (61%)
discr (1000)	0.400 (42%)	0.458 (49%)	0.384 (47%)	0.414 (55%)	0.346 (52%)
discr (1500)	0.397 (37%)	0.458 (44%)	0.384 (42%)	0.398 (50%)	0.348 (46%)
discr (2000)	0.400 (34%)	0.468 (41%)	0.375 (39%)	0.400 (47%)	0.352 (42%)
discr (2500)	0.409 (31%)	0.475'(38%)	0.376 (36%)	0.399 (44%)	0.380 (39%)
discr (5000)	0.410'(24%)	0.498'(29%)	0.396 (28%)	0.408'(36%)	0.431 (30%)
χ^2 (50)	0.420 (85%)	0.451 (89%)	0.452 (88%)	0.384 (90%)	0.347 (96%)
χ^2 (100)	0.432 (85%)	0.457 (84%)	0.449 (83%)	0.389 (86%)	0.349 (92%)
χ^2 (250)	0.416 (83%)	0.459 (74%)	0.444 (73%)	0.389 (78%)	0.349 (83%)
χ^2 (500)	0.425 (79%)	0.455 (60%)	0.433 (62%)	0.392 (67%)	0.352 (70%)
χ^2 (1000)	0.419 (77%)	0.462 (54%)	0.441 (57%)	0.401 (61%)	0.347 (63%)
χ^2 (1500)	0.432 (75%)	0.468 (49%)	0.461 (53%)	0.401 (57%)	0.347 (58%)
χ^2 (2000)	0.439 (72%)	0.480 (44%)	0.461 (49%)	0.402 (53%)	0.347 (53%)
χ^2 (2500)	0.423 (70%)	0.483 (41%)	0.440 (46%)	0.406 (49%)	0.352 (49%)
χ^2 (5000)	0.455 (57%)	0.501'(30%)	0.429 (36%)	0.418'(39%)	0.352 (36%)
E (50)	0.419 (94%)	0.452 (95%)	0.452 (95%)	0.392 (96%)	0.349 (95%)
E (100)	0.425 (87%)	0.452 (83%)	0.445 (83%)	0.396 (88%)	0.350 (86%)
E (250)	0.407 (65%)	0.463 (66%)	0.436 (66%)	0.394 (74%)	0.351 (73%)
E (500)	0.412 (49%)	0.459 (56%)	0.429 (56%)	0.403 (64%)	0.346 (62%)
E (1000)	0.415 (37%)	0.462 (47%)	0.413 (49%)	0.406 (55%)	0.344 (53%)
E (1500)	0.413'(32%)	0.462 (42%)	0.413 (44%)	0.398 (50%)	0.346 (47%)
E (2000)	0.428'(29%)	0.470'(39%)	0.417 (41%)	0.397'(46%)	0.347 (43%)
E (2500)	0.433'(27%)	0.475'(37%)	0.407 (38%)	0.396'(44%)	0.346 (40%)
E (5000)	0.437'(22%)	0.509'(29%)	0.454 (30%)	0.407'(37%)	0.421 (31%)
E' (50)	0.417 (89%)	0.449 (93%)	0.456 (92%)	0.387 (94%)	0.349 (98%)
E' (100)	0.417 (89%)	0.451 (93%)	0.454 (92%)	0.387 (94%)	0.349 (98%)
E' (250)	0.422 (88%)	0.455 (90%)	0.454 (89%)	0.393 (92%)	0.350 (96%)
E' (500)	0.420 (88%)	0.453 (89%)	0.451 (88%)	0.388 (90%)	0.347 (94%)
E' (1000)	0.420 (87%)	0.449 (86%)	0.449 (86%)	0.386 (88%)	0.346 (92%)

continued on next page

continued from previous page

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
E' (1500)	0.421 (87%)	0.449 (83%)	0.447 (83%)	0.392 (85%)	0.347 (88%)
E' (2000)	0.417 (86%)	0.459 (79%)	0.440 (79%)	0.395 (82%)	0.348 (83%)
E' (2500)	0.414 (86%)	0.462 (76%)	0.434 (76%)	0.392 (78%)	0.346 (79%)
E' (5000)	0.424 (84%)	0.477 (66%)	0.424 (67%)	0.399 (69%)	0.349 (70%)
E" (50)	0.414 (79%)	0.455 (76%)	0.445 (75%)	0.391 (81%)	0.349 (81%)
E" (100)	0.414 (60%)	0.459 (66%)	0.435 (63%)	0.384 (73%)	0.350 (76%)
E" (250)	0.403 (50%)	0.458 (59%)	0.419 (57%)	0.393 (65%)	0.350 (67%)
E" (500)	0.406 (42%)	0.459 (52%)	0.401 (51%)	0.400 (59%)	0.348 (58%)
E" (1000)	0.407 (35%)	0.465 (45%)	0.373 (44%)	0.403 (52%)	0.348 (51%)
E" (1500)	0.421 (31%)	0.470 (41%)	0.374 (39%)	0.395 (48%)	0.347 (46%)
E" (2000)	0.405 (29%)	0.475 (38%)	0.382 (36%)	0.401'(45%)	0.345 (43%)
E" (2500)	0.420 (27%)	0.474'(36%)	0.393 (34%)	0.403'(43%)	0.346 (40%)
E" (5000)	0.445'(22%)	0.494'(30%)	0.395 (28%)	0.404'(37%)	0.389 (32%)
KL (50)	0.425 (70%)	0.459 (76%)	0.449 (74%)	0.386 (80%)	0.352 (84%)
KL (100)	0.422 (66%)	0.461 (71%)	0.445 (68%)	0.386 (76%)	0.350 (78%)
KL (250)	0.436 (60%)	0.462 (63%)	0.424 (61%)	0.391 (68%)	0.352 (69%)
KL (500)	0.444 (54%)	0.459 (56%)	0.413 (54%)	0.401 (61%)	0.346 (61%)
KL (1000)	0.423 (46%)	0.458 (48%)	0.389 (47%)	0.404 (54%)	0.346 (51%)
KL (1500)	0.415 (39%)	0.470'(43%)	0.387 (42%)	0.409 (50%)	0.346 (46%)
KL (2000)	0.403 (36%)	0.470'(40%)	0.374 (39%)	0.402 (47%)	0.346 (42%)
KL (2500)	0.409 (33%)	0.478'(37%)	0.395 (36%)	0.405 (44%)	0.387 (39%)
KL (5000)	0.443 (26%)	0.506'(29%)	0.440 (28%)	0.409'(36%)	0.430 (30%)
WKL (50)	0.428 (71%)	0.460 (77%)	0.449 (74%)	0.392 (80%)	0.350 (85%)
WKL (100)	0.417 (67%)	0.454 (71%)	0.445 (69%)	0.385 (76%)	0.350 (78%)
WKL (250)	0.409 (58%)	0.457 (61%)	0.421 (59%)	0.399 (66%)	0.350 (67%)
WKL (500)	0.412 (51%)	0.458 (53%)	0.398 (52%)	0.398 (59%)	0.349 (58%)
WKL (1000)	0.412 (41%)	0.459 (45%)	0.379 (43%)	0.413 (51%)	0.352 (48%)
WKL (1500)	0.407 (35%)	0.471'(40%)	0.389 (38%)	0.410'(47%)	0.351 (42%)
WKL (2000)	0.417 (32%)	0.470'(37%)	0.390 (34%)	0.423'(43%)	0.411 (38%)
WKL (2500)	0.421 (29%)	0.485'(34%)	0.393 (31%)	0.425'(41%)	0.440 (35%)
WKL (5000)	0.461'(23%)	0.527'(27%)	0.416 (24%)	0.442'(34%)	0.437 (27%)
BOL_{stop}	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)

Table D.14: **Cross-validation of different stopword discrimination measures (union).** Stopword lists are derived by *unifying* the stopword candidate lists of the other four data sets. [Section 5.4.2.2]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
df (50)	0.420 (76%)	0.455 (84%)	0.455 (81%)	0.388 (86%)	0.349 (91%)
df (100)	0.426 (72%)	0.449 (80%)	0.451 (77%)	0.386 (83%)	0.350 (87%)
df (250)	0.421 (68%)	0.463 (74%)	0.446 (71%)	0.386 (79%)	0.350 (82%)
df (500)	0.416 (64%)	0.459 (69%)	0.441 (67%)	0.388 (75%)	0.347 (77%)
df (1000)	0.422 (57%)	0.460 (62%)	0.436 (60%)	0.387 (68%)	0.347 (70%)
df (1500)	0.423 (51%)	0.458 (58%)	0.428 (57%)	0.391 (65%)	0.351 (65%)
df (2000)	0.425 (47%)	0.463 (54%)	0.425 (54%)	0.392 (61%)	0.354 (61%)
df (2500)	0.410 (44%)	0.463 (52%)	0.421 (51%)	0.399 (59%)	0.347 (58%)
df (5000)	0.422 (36%)	0.473 (44%)	0.430 (43%)	0.414 (51%)	0.347 (49%)
discr (50)	0.416 (84%)	0.449 (90%)	0.456 (87%)	0.388 (92%)	0.350 (95%)
discr (100)	0.412 (80%)	0.452 (86%)	0.455 (83%)	0.386 (89%)	0.349 (93%)
discr (250)	0.417 (75%)	0.450 (82%)	0.450 (78%)	0.388 (85%)	0.353 (89%)
discr (500)	0.416 (71%)	0.455 (77%)	0.447 (74%)	0.389 (82%)	0.350 (85%)
discr (1000)	0.422 (68%)	0.461 (73%)	0.443 (69%)	0.389 (78%)	0.351 (80%)
discr (1500)	0.426 (66%)	0.457 (69%)	0.441 (65%)	0.390 (75%)	0.347 (76%)
discr (2000)	0.425 (64%)	0.462 (67%)	0.436 (63%)	0.387 (72%)	0.348 (73%)
discr (2500)	0.425 (62%)	0.462 (65%)	0.429 (60%)	0.397 (70%)	0.350 (70%)
discr (5000)	0.439 (55%)	0.465 (59%)	0.412 (53%)	0.402 (63%)	0.351 (62%)
χ^2 (50)	———	———	———	———	———
χ^2 (100)	———	———	———	———	———
χ^2 (250)	———	0.453 (99%)	———	0.389 (99%)	———
χ^2 (500)	———	0.451 (98%)	———	0.386 (98%)	———
χ^2 (1000)	———	0.451 (98%)	———	0.386 (98%)	———
χ^2 (1500)	0.418 (100%)	0.449 (98%)	———	0.389 (98%)	0.351 (100%)
χ^2 (2000)	0.418 (100%)	0.449 (98%)	———	0.389 (98%)	0.350 (100%)
χ^2 (2500)	0.418 (100%)	0.448 (97%)	0.457 (99%)	0.392 (98%)	0.352 (100%)
χ^2 (5000)	0.418 (100%)	0.452 (93%)	0.457 (96%)	0.385 (94%)	0.350 (98%)
E (250)	0.416 (100%)	0.450 (98%)	0.456 (98%)	———	0.351 (99%)
E (500)	0.414 (99%)	0.446 (95%)	0.454 (95%)	0.390 (100%)	0.348 (96%)
E (1000)	0.414 (91%)	0.453 (85%)	0.447 (89%)	0.388 (94%)	0.349 (92%)
E (1500)	0.420 (82%)	0.458 (78%)	0.445 (81%)	0.389 (88%)	0.349 (88%)
E (2000)	0.410 (71%)	0.459 (73%)	0.442 (74%)	0.391 (82%)	0.349 (82%)
E (2500)	0.406 (68%)	0.454 (70%)	0.438 (70%)	0.391 (79%)	0.350 (80%)
E (5000)	0.414 (54%)	0.463 (58%)	0.416 (55%)	0.404 (65%)	0.354 (64%)
E' (50)	———	———	———	———	———
E' (100)	———	———	———	———	———
E' (250)	———	———	———	———	———
E' (500)	———	———	———	———	———
E' (1000)	———	———	———	———	———
E' (1500)	———	———	———	———	———
E' (2000)	———	———	———	———	———

continued on next page

continued from previous page

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
E' (2500)	————	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
E' (5000)	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.389 (100%)	0.349 (100%)
E" (50)	————	0.449 (100%)	————	————	0.350 (100%)
E" (100)	0.418 (97%)	0.452 (97%)	0.456 (98%)	0.391 (96%)	0.350 (96%)
E" (250)	0.425 (87%)	0.454 (89%)	0.449 (89%)	0.387 (92%)	0.349 (87%)
E" (500)	0.412 (67%)	0.460 (71%)	0.446 (70%)	0.390 (77%)	0.348 (80%)
E" (1000)	0.417 (63%)	0.459 (66%)	0.438 (65%)	0.390 (74%)	0.347 (76%)
E" (1500)	0.408 (60%)	0.462 (64%)	0.429 (62%)	0.389 (71%)	0.346 (72%)
E" (2000)	0.413 (58%)	0.466 (62%)	0.425 (60%)	0.385 (70%)	0.346 (71%)
E" (2500)	0.412 (56%)	0.461 (60%)	0.420 (58%)	0.392 (68%)	0.352 (69%)
E" (5000)	0.421 (52%)	0.464 (55%)	0.417 (53%)	0.390 (63%)	0.349 (63%)
KL (50)	0.419 (83%)	0.451 (89%)	0.457 (86%)	0.387 (91%)	0.347 (95%)
KL (100)	0.423 (80%)	0.447 (86%)	0.454 (81%)	0.391 (88%)	0.349 (92%)
KL (250)	0.420 (75%)	0.454 (79%)	0.448 (74%)	0.391 (84%)	0.353 (87%)
KL (500)	0.448 (70%)	0.460 (75%)	0.447 (71%)	0.394 (79%)	0.350 (82%)
KL (1000)	0.422 (66%)	0.458 (69%)	0.440 (65%)	0.394 (74%)	0.347 (75%)
KL (1500)	0.435 (64%)	0.460 (66%)	0.428 (61%)	0.391 (71%)	0.346 (71%)
KL (2000)	0.425 (61%)	0.460 (64%)	0.421 (59%)	0.392 (68%)	0.350 (68%)
KL (2500)	0.418 (59%)	0.468 (62%)	0.422 (56%)	0.394 (67%)	0.349 (65%)
KL (5000)	0.421 (52%)	0.464 (54%)	0.392 (50%)	0.400 (59%)	0.353 (57%)
WKL (50)	0.421 (78%)	0.446 (85%)	0.455 (83%)	0.387 (87%)	0.348 (91%)
WKL (100)	0.426 (73%)	0.456 (80%)	0.451 (77%)	0.384 (83%)	0.348 (88%)
WKL (250)	0.426 (69%)	0.454 (73%)	0.447 (70%)	0.387 (78%)	0.347 (81%)
WKL (500)	0.418 (64%)	0.464 (69%)	0.443 (67%)	0.388 (74%)	0.347 (76%)
WKL (1000)	0.409 (57%)	0.461 (62%)	0.423 (60%)	0.388 (67%)	0.347 (69%)
WKL (1500)	0.416 (52%)	0.458 (58%)	0.420 (56%)	0.395 (64%)	0.349 (64%)
WKL (2000)	0.404 (49%)	0.464 (55%)	0.405 (53%)	0.396 (61%)	0.349 (60%)
WKL (2500)	0.388 (45%)	0.465 (52%)	0.401 (51%)	0.402 (59%)	0.352 (57%)
WKL (5000)	0.413 (36%)	0.470 (44%)	0.409 (43%)	0.403 (51%)	0.345 (48%)
BOL_{stop}	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)

Table D.15: **Cross-validation of different stopword discrimination measures (intersection).** Stopword lists are derived by *intersecting* the stopword candidate lists of the other four data sets. [Section 5.4.2.2]

AMAZON *subset size*	*selected labels*	BOL	BOL_{stop}
5	ARCH, BUSI, HIST, KUNS, RATG	0.511	0.451
	BELL, COMP, INGE, LIFE, REIS	0.276	0.270
	BIOC, EROS, KIND, MATH, RELI	0.283	0.280
	BIOG, GERM, KULT, PUBL, SCIF	0.496	0.480
	BUSI, KIND, PUBL, RELI, SPOR	0.285	0.271
7	ARCH, BIOG, EROS, INGE, KUNS, PUBL, RELI	0.390	0.379
	BELL, BUSI, GERM, KIND, LIFE, RATG, SCIF	0.394	0.392
	BELL, COMP, EROS, KIND, MATH, PUBL, SCIF	0.320	0.315
	BIOC, COMP, HIST, KULT, MATH, REIS, SPOR	0.549	0.541
	BIOG, GERM, KULT, KUNS, RATG, RELI, SPOR	0.553	0.538
9	ARCH, BIOC, BUSI, EROS, HIST, KIND, KUNS, MATH, RATG	0.356	0.352
	ARCH, BIOG, BUSI, GERM, HIST, KULT, KUNS, PUBL, RELI	0.547	0.528
	BELL, BIOC, COMP, EROS, INGE, KIND, LIFE, MATH, SCIF	0.376	0.400
	BELL, BIOG, COMP, GERM, INGE, KULT, LIFE, PUBL, REIS	0.438	0.427
	BIOC, COMP, GERM, HIST, RATG, REIS, RELI, SCIF, SPOR	0.495	0.475
11	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND	0.451	0.440
	ARCH, BIOC, BUSI, EROS, HIST, KIND, KUNS, MATH, RATG, RELI, SPOR	0.403	0.393
	BELL, BIOG, COMP, GERM, INGE, KULT, LIFE, PUBL, REIS, SCIF, SPOR	0.424	0.424
	BIOC, BIOG, BUSI, HIST, INGE, KIND, LIFE, MATH, RATG, RELI, SCIF	0.420	0.424
	KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.362	0.357
13	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS	0.461	0.454
	ARCH, BELL, BIOC, BUSI, EROS, HIST, KIND, KUNS, MATH, RATG, RELI, SCIF, SPOR	0.403	0.402
	ARCH, BELL, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.388	0.392
	BELL, BIOC, BIOG, COMP, GERM, INGE, KULT, LIFE, PUBL, RATG, REIS, SCIF, SPOR	0.463	0.459
	BIOC, BIOG, BUSI, EROS, HIST, INGE, KIND, LIFE, MATH, RATG, REIS, RELI, SCIF	0.416	0.413
15	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH	0.445	0.449
	ARCH, BELL, BIOC, BIOG, BUSI, EROS, HIST, KIND, KUNS, MATH, RATG, REIS, RELI, SCIF, SPOR	0.428	0.440
	ARCH, BELL, BIOC, BIOG, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.432	0.432

continued on next page

continued from previous page

AMAZON *subset size*	*selected labels*	BOL	BOL_{stop}
	BELL, BIOC, BIOG, BUSI, COMP, GERM, INGE, KULT, LIFE, MATH, PUBL, RATG, REIS, SCIF, SPOR	0.461	0.473
	BIOC, BIOG, BUSI, EROS, GERM, HIST, INGE, KIND, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF	0.420	0.416
17	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG	0.456	0.454
	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, HIST, KIND, KUNS, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.422	0.451
	ARCH, BELL, BIOC, BIOG, BUSI, EROS, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.429	0.431
	BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, INGE, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, SCIF, SPOR	0.457	0.459
	BIOC, BIOG, BUSI, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF	0.438	0.431
19	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI	0.439	0.461
	ARCH, BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, KIND, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.437	0.457
	ARCH, BELL, BIOC, BIOG, BUSI, EROS, GERM, HIST, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF, SPOR	0.449	0.450
	ARCH, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, RELI, SCIF	0.443	0.438
	BELL, BIOC, BIOG, BUSI, COMP, EROS, GERM, HIST, INGE, KIND, KULT, KUNS, LIFE, MATH, PUBL, RATG, REIS, SCIF, SPOR	0.459	0.456

Table D.16: **Subset experiments for stopword removal (AMAZON).** [Section 5.6]

WIKI *subset size*	*selected labels*	BOL	BOL_{stop}
5	ASTR, FREI, KUNS, MEDZ, SEXU	0.288	0.274
	BIOL, HIST, LIFE, ORGA, SOZI	0.386	0.362
	BUSI, INFO, LITE, PHYS, SPOR	0.246	0.243
	CHEM, JURA, MATH, RELI, SPRA	0.304	0.337
	FREI, LITE, RELI, SPOR, TECH	0.372	0.372

continued on next page

continued from previous page

WIKI *subset size*	*selected labels*	BOL	BOL_{stop}
7	ASTR, CHEM, INFO, LIFE, MEDZ, RELI, SPOR	0.193	0.199
	BIOL, FREI, JURA, LITE, ORGA, SEXU, SPRA	0.325	0.298
	BIOL, HIST, INFO, LITE, PHYS, RELI, SPRA	0.422	0.435
	BUSI, HIST, KUNS, MATH, PHYS, SOZI, TECH	0.500	0.416
	CHEM, JURA, MATH, MEDZ, SEXU, SPOR, TECH	0.195	0.189
9	ASTR, BUSI, FREI, INFO, KUNS, LITE, MEDZ, PHYS, SEXU	0.327	0.320
	ASTR, CHEM, FREI, JURA, KUNS, MATH, MEDZ, RELI, SPOR	0.272	0.270
	BIOL, BUSI, HIST, INFO, LIFE, LITE, ORGA, PHYS, SPRA	0.390	0.389
	BIOL, CHEM, HIST, JURA, LIFE, MATH, ORGA, RELI, SOZI	0.337	0.314
	BUSI, HIST, JURA, KUNS, SEXU, SOZI, SPOR, SPRA, TECH	0.430	0.410
11	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE	0.386	0.394
	ASTR, BUSI, FREI, INFO, KUNS, LITE, MEDZ, PHYS, SEXU, SPOR, TECH	0.375	0.371
	BIOL, CHEM, HIST, JURA, LIFE, MATH, ORGA, RELI, SOZI, SPRA, TECH	0.370	0.366
	BUSI, CHEM, FREI, KUNS, LIFE, LITE, ORGA, PHYS, SEXU, SPOR, SPRA	0.373	0.353
	LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.400	0.388
13	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ	0.386	0.392
	ASTR, BIOL, BUSI, FREI, INFO, KUNS, LITE, MEDZ, PHYS, SEXU, SPOR, SPRA, TECH	0.388	0.402
	ASTR, BIOL, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.364	0.338
	BIOL, BUSI, CHEM, HIST, JURA, LIFE, MATH, ORGA, RELI, SEXU, SOZI, SPRA, TECH	0.406	0.393
	BUSI, CHEM, FREI, INFO, KUNS, LIFE, LITE, ORGA, PHYS, SEXU, SOZI, SPOR, SPRA	0.379	0.349
15	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS	0.389	0.382
	ASTR, BIOL, BUSI, CHEM, FREI, INFO, KUNS, LITE, MEDZ, PHYS, SEXU, SOZI, SPOR, SPRA, TECH	0.387	0.405
	ASTR, BIOL, BUSI, CHEM, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.384	0.382
	BIOL, BUSI, CHEM, FREI, HIST, JURA, LIFE, MATH, ORGA, PHYS, RELI, SEXU, SOZI, SPRA, TECH	0.384	0.398
	BUSI, CHEM, FREI, INFO, JURA, KUNS, LIFE, LITE, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA	0.375	0.371
17	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU	0.404	0.401

continued on next page

continued from previous page

WIKI *subset size*	*selected labels*	BOL	BOL_{stop}
	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, KUNS, LITE, MEDZ, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.438	0.428
	ASTR, BIOL, BUSI, CHEM, FREI, INFO, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.381	0.384
	BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, LIFE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPRA, TECH	0.390	0.393
	BUSI, CHEM, FREI, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA	0.372	0.348
19	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR	0.397	0.377
	ASTR, BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LITE, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.416	0.417
	ASTR, BIOL, BUSI, CHEM, FREI, INFO, JURA, KUNS, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA, TECH	0.389	0.397
	ASTR, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPOR, SPRA	0.430	0.412
	BIOL, BUSI, CHEM, FREI, HIST, INFO, JURA, KUNS, LIFE, LITE, MATH, MEDZ, ORGA, PHYS, RELI, SEXU, SOZI, SPRA, TECH	0.414	0.399

Table D.17: **Subset experiments for stopword removal (WIKI).** [Section 5.6]

POS selection	SPRINGER	AMAZON	SDA	WIKI	NZZ
ADJ[a]	0.671'(17%)	0.630'(16%)	0.647'(11%)	0.651'(14%)	0.509 (17%)
VRB[b]	0.904'(12%)	0.695'(15%)	0.716'(17%)	0.725'(13%)	0.650 (16%)
SUB	0.440 (37%)	0.479 (31%)	0.440 (35%)	0.408 (36%)	0.367 (37%)
SUB, ADJ	**0.403 (54%)**	**0.457 (47%)**	**0.394 (46%)**	**0.405 (50%)**	**0.355 (54%)**
SUB, VRB	0.448 (50%)	0.467 (46%)	0.486 (52%)	0.403 (50%)	0.389 (53%)
SUB, APP	0.447 (47%)	0.470 (39%)	0.456 (46%)	0.404 (43%)	0.365 (43%)
SUB, NUM	0.442 (38%)	0.478 (32%)	0.457 (36%)	0.401 (37%)	0.367 (38%)
SUB, UNK	0.441 (38%)	0.474 (32%)	0.444 (35%)	0.401 (37%)	0.369 (37%)
SUB, NAM	0.443 (39%)	0.474 (36%)	0.413 (40%)	0.399 (44%)	0.351 (41%)
SUB, NAM$_{all}$	0.450 (39%)	0.478 (39%)	0.406 (41%)	0.403 (48%)	0.349 (42%)
SUB, NAM, UNK	0.444 (40%)	0.474 (37%)	0.393 (40%)	0.397 (45%)	0.349 (41%)
SUB, NAM, UNK, NUM	0.439 (40%)	0.473 (38%)	0.398 (41%)	0.404 (46%)	0.350 (42%)
SUB, UNK, NAM$_{all}$	0.446 (40%)	0.477 (40%)	0.404 (42%)	0.406 (49%)	0.350 (43%)
SUB, ADJ, NAM$_{all}$	**0.402 (57%)**	**0.468 (55%)**	**0.416 (53%)**	**0.393 (61%)**	**0.348 (59%)**
SUB, ADJ, UNK	0.417 (55%)	0.454 (48%)	0.423 (47%)	0.416 (50%)	0.355 (54%)
SUB, ADJ, UNK, NAM$_{all}$	0.398 (58%)	0.468 (56%)	0.415 (53%)	0.407 (62%)	0.355 (60%)
SUB, ADJ, VRB	0.412 (67%)	0.454 (62%)	0.480 (63%)	0.402 (63%)	0.364 (69%)
SUB, ADJ, VRB, NAM$_{all}$	0.410 (69%)	0.461 (70%)	0.436 (69%)	0.388 (75%)	0.350 (75%)
SUB, ADJ, VRB, UNK, NAM$_{all}$	0.411 (70%)	0.460 (71%)	0.434 (70%)	0.390 (75%)	0.350 (76%)
SUB, ADJ, VRB, ADV, APP, NAM$_{all}$, NUM, UNK	0.414 (84%)	0.457 (86%)	0.448 (87%)	0.388 (89%)	0.347 (90%)
all	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
BOL$_{\text{stop}}$	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)
BOW$_{\text{stop, stem}}$ (Baseline 1)	0.410 (62%)	0.468 (61%)	0.431 (61%)	0.416 (68%)	0.348 (69%)

[a] 4, 307, 116, 747 resp. 0 documents were omitted because of a lack of shared adjectives.
[b] 16, 679, 10, 120 resp. 0 documents were omitted because of a lack of shared verbs.

Table D.18: **POS-based feature selection.** Percentages indicate the number of features surviving the selection step. [Section 5.5]

POS classes	*weight*	SPRINGER	AMAZON	SDA	WIKI	NZZ
—	—	0.421[0.015]	0.448[0.007]	0.456[0.001]	0.388[0.007]	0.349[0.005]
SUB	1.5	0.419[0.017]	0.451[0.006]	0.439[0.001]	0.394[0.015]	0.344[0.001]
SUB	2	0.422[0.017]	0.455[0.009]	0.418[0.023]	0.396[0.019]	0.345[0.003]
SUB	3	0.432[0.012]	0.461[0.004]	0.433[0.030]	0.404[0.015]	0.351[0.003]
NAM_{all}	1.5	0.431[0.010]	0.469[0.012]	0.445[0.001]	0.401[0.009]	0.359[0.008]
NAM_{all}	2	0.421[0.013]	0.490[0.006]	0.442[0.000]	0.419[0.011]	0.373[0.003]
NAM_{all}	3	0.445[0.012]	0.508[0.007]	0.440[0.002]	0.433[0.009]	0.396[0.015]
SUB, ADJ	1.5	0.415[0.013]	0.446[0.005]	0.446[0.002]	0.387[0.018]	0.344[0.001]
SUB, ADJ	2	0.414[0.011]	0.447[0.005]	0.451[0.009]	0.390[0.017]	0.347[0.002]
SUB, ADJ	3	0.419[0.014]	0.449[0.007]	0.443[0.026]	0.400[0.017]	0.350[0.004]
SUB, NAM_{all}	1.5	0.422[0.019]	0.462[0.008]	0.437[0.000]	0.393[0.010]	0.343[0.005]
SUB, NAM_{all}	2	0.418[0.012]	0.461[0.005]	0.426[0.003]	0.398[0.009]	0.343[0.003]
SUB, NAM_{all}	3	0.431[0.018]	0.462[0.004]	0.417[0.002]	0.397[0.016]	0.341[0.001]
SUB, ADJ, NAM_{all}	1.5	0.408[0.007]	0.455[0.010]	0.439[0.001]	0.388[0.006]	0.342[0.001]
SUB, ADJ, NAM_{all}	2	0.414[0.007]	0.456[0.008]	0.435[0.001]	0.390[0.008]	0.342[0.001]
SUB, ADJ, NAM_{all}	3	0.416[0.012]	0.460[0.010]	0.431[0.001]	0.396[0.017]	0.341[0.001]

Table D.19: **Lending extra weight to chosen POS categories.** [Section 5.7.1]

stopword weight γ	SPRINGER	AMAZON	SDA	WIKI	NZZ
1	0.421[0.015]	0.448[0.007]	0.456[0.001]	0.388[0.007]	0.349[0.005]
0.9	0.417[0.012]	0.453[0.007]	0.450[0.002]	0.386[0.009]	0.349[0.004]
0.75	0.413[0.012]	0.458[0.008]	0.443[0.001]	0.390[0.011]	0.348[0.005]
0.5	0.417[0.018]	0.461[0.004]	0.424[0.001]	0.416[0.023]	0.347[0.004]
0.25	0.412[0.016]	0.465[0.009]	0.422[0.002]	0.411[0.020]	0.347[0.004]
0.1	0.425[0.016]	0.460[0.003]	0.422[0.000]	0.408[0.012]	0.349[0.004]
0	0.422[0.018]	0.463[0.005]	0.421[0.001]	0.408[0.019]	0.349[0.005]

Table D.20: **Stopword weighting instead of elimination.** From treating stopwords like all other words ($\gamma = 1$) to complete stopword removal ($\gamma = 0$). [Section 5.7.2]

n	*weight*	SPRINGER	AMAZON	SDA	WIKI	NZZ
0	—	0.421[0.015]	0.448[0.007]	0.456[0.001]	0.388[0.007]	0.349[0.005]
5	2	0.430[0.012]	0.459[0.009]	0.452[0.005]	0.399[0.021]	0.349[0.005]
5	3	0.424[0.015]	0.467[0.007]	0.450[0.004]	0.419[0.009]	0.347[0.005]
10	2	0.423[0.015]	0.461[0.009]	0.446[0.003]	0.384[0.015]	0.347[0.004]
10	3	0.435[0.016]	0.465[0.003]	0.448[0.004]	0.397[0.014]	0.346[0.003]

Table D.21: **Lending extra weight to the first n nouns.** [Section 5.7.3]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
S1: BOW	0.476	0.460	0.474	0.392	0.348
S2: BOW_{stem}	0.408	0.458	0.463	0.386	0.346
S3: BOW_{stop}	0.449	0.466	0.436	0.416	0.346
S4: $\text{BOW}_{\text{stop, stem}}$	0.410	0.468	0.431	0.416	0.348
S5: *best of S1–S4*	0.408	0.458	0.431	0.386	0.346
L1: BOL	0.421	0.448	0.456	0.388	0.349
L2: BOL_{stop}	0.422	0.463	0.421	0.408	0.349
L3: POS selection: SUB, ADJ, NAM_{all}	0.402	0.468	0.416	0.393	0.348
L4: POS weighting (1.5): SUB, ADJ	0.415	0.446	0.446	0.387	0.344
L5: POS weighting (1.5): SUB, ADJ, NAM_{all}	0.408	0.455	0.439	0.388	0.342
L6: *best of L1–L5*	0.402	0.446	0.416	0.387	0.342
M1: stopwords with E'' (Union, $\alpha = 250$)	0.403	0.458	0.419	0.393	0.350
M2: local pruning (0–10%)	0.416	0.460	0.448	0.394	0.353

Table D.22: **Linguistic and statistical reduction methods in comparison.** [Section 5.8]

D.3 Experiments in Chapter 6 (Matrix Enhancement)

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOW_{stop}	0.449 (100%)	0.466 (100%)	0.436 (100%)	0.416 (100%)	0.346 (100%)
$BOW_{stop,\ stem}$	0.410 (97%)	0.468 (96%)	0.431 (96%)	0.416 (94%)	0.348 (94%)
$BOW_{stop,\ split,\ stem}$	0.424 (102%)	0.466 (99%)	0.416 (101%)	0.407 (98%)	0.347 (97%)
$BOW_{stop,\ *split,\ stem}$	0.417 (98%)	0.470 (97%)	0.419 (97%)	0.409 (95%)	0.348 (94%)
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
$BOL_{stop,\ split}$	0.435 (104%)	0.464 (103%)	0.412 (105%)	0.406 (104%)	0.350 (102%)
$BOL_{stop,\ *split}$	0.425 (101%)	0.462 (101%)	0.415 (101%)	0.402 (101%)	0.351 (100%)

Table D.23: **Splitting mechanical compounds.** BOW and BOL extended by splitting "mechanical" compounds (A-B and A/B features). Stars refer to experiments in which the compound token was *discarded* after analysis and replacement by the constituents. [Section 6.1.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
$BOL_{stop,\ split}$	0.360 (138%)	0.450 (122%)	0.373 (131%)	0.384 (124%)	0.346 (124%)
$BOL_{stop,\ *split}$	0.355 (115%)	0.467 (108%)	0.407 (111%)	0.400 (106%)	0.407 (106%)

Table D.24: **Splitting all compounds** (after stopword removal). [Section 6.1.2]

		SPRINGER	AMAZON	SDA	WIKI	NZZ
	BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
			split all			
—	$BOL_{stop,\ split}$	0.360 (138%)	0.450 (122%)	0.373 (131%)	0.384 (124%)	0.346 (124%)
			l: minimum word length			
l10	word length >= 10	0.358 (134%)	0.459 (118%)	0.371 (126%)	0.386 (120%)	0.344 (121%)
l15	word length >= 15	0.363 (119%)	0.463 (107%)	0.410 (112%)	0.389 (109%)	0.354 (109%)
l20	word length >= 20	0.383 (106%)	0.461 (102%)	0.418 (103%)	0.409 (102%)	0.350 (102%)
l25	word length >= 25	0.419 (101%)	0.462 (100%)	0.421 (101%)	0.413 (101%)	0.348 (100%)
			d: maximum document frequency			
d1	doc. freq. <= 1	0.359 (115%)	0.459 (104%)	0.420 (104%)	0.408 (106%)	0.344 (104%)
d5	doc. freq. <= 5	0.350 (124%)	0.462 (108%)	0.413 (108%)	0.389 (111%)	0.343 (108%)
d15	doc. freq. <= 15	**0.347** (129%)	0.462 (112%)	0.393 (112%)	0.392 (115%)	**0.342** (111%)
d25	doc. freq. <= 25	0.351 (132%)	0.462 (113%)	0.381 (114%)	0.389 (116%)	0.344 (113%)
d50	doc. freq. <= 50	0.353 (134%)	0.454 (116%)	0.375 (118%)	0.389 (119%)	0.346 (116%)
d75	doc. freq. <= 75	0.358 (135%)	0.457 (117%)	0.369 (120%)	0.394 (120%)	0.346 (117%)
d100	doc. freq. <= 100	0.360 (135%)	0.453 (117%)	0.371 (121%)	0.386 (120%)	0.346 (118%)
d150	doc. freq. <= 150	0.363 (136%)	0.449 (118%)	**0.368** (123%)	0.390 (121%)	0.346 (120%)
d200	doc. freq. <= 200	0.360 (137%)	**0.445** (119%)	0.370 (124%)	0.384 (122%)	0.346 (120%)
d250	doc. freq. <= 250	0.360 (137%)	0.450 (119%)	0.370 (125%)	0.389 (122%)	0.346 (121%)
d500	doc. freq. <= 500	0.359 (138%)	0.449 (120%)	0.373 (127%)	0.390 (123%)	0.346 (122%)
			c: compound POS restriction			
cN	POS = SUB	0.358 (133%)	0.453 (119%)	0.375 (128%)	0.389 (122%)	0.346 (121%)
cNA	POS = SUB or ADJ	0.359 (137%)	0.451 (121%)	0.375 (130%)	0.388 (124%)	0.347 (124%)
			s: constituent POS restriction			
sSN	compound = [all]+N	0.360 (135%)	0.451 (120%)	0.376 (128%)	0.392 (122%)	0.345 (122%)
sSN-MN	compound = N+N	0.356 (130%)	0.459 (116%)	0.373 (123%)	0.391 (118%)	0.345 (118%)
sMN	compound = N+[all]	0.358 (134%)	0.452 (118%)	0.377 (126%)	0.390 (120%)	0.345 (120%)
			r: lexical restriction			
rT3	Skip 1,000 most frequent terms	0.359 (138%)	0.450 (122%)	0.374 (130%)	0.396 (124%)	0.346 (124%)
rT4	Skip 10,000 most frequent terms	0.357 (135%)	0.450 (120%)	0.383 (127%)	0.397 (122%)	0.346 (122%)
rC	Only from corpus	0.364 (130%)	0.460 (118%)	0.380 (125%)	0.386 (120%)	0.346 (120%)
			x/X: multi-part constituents			
x	all combinations	0.355 (139%)	0.455 (122%)	0.385 (131%)	0.390 (125%)	0.344 (125%)
xrC	all combinations in corpus	0.356 (130%)	0.456 (118%)	0.384 (125%)	0.391 (120%)	0.344 (120%)
X	outer combinations	0.357 (138%)	0.450 (122%)	0.378 (131%)	0.396 (125%)	0.344 (125%)
XrC	outer combinations in corpus	0.361 (130%)	0.458 (118%)	0.384 (125%)	**0.383** (120%)	0.344 (120%)

Table D.25: **Modifications to compound splitting.** The best result for each data set is printed in bold digits. On the whole the splitting baseline $BOL_{stop,\ split}$ ("split all") was difficult to beat. [Section 6.1.3]

subset size (labels)		subset 1	subset 2	subset 3	subset 4	subset 5	eval
5	BOL	0.511	0.276	0.283	0.496	0.285	0
5	BOL + split	0.433	0.274	0.276	0.492	0.274	5
7	BOL	0.390	0.394	0.320	0.549	0.553	3
7	BOL + split	0.469	0.409	0.328	0.544	0.548	2
9	BOL	0.356	0.547	0.376	0.438	0.495	0
9	BOL + split	0.353	0.541	0.369	0.427	0.459	5
11	BOL	0.451	0.403	0.424	0.420	0.362	1
11	BOL + split	0.430	0.383	0.423	0.429	0.350	4
13	BOL	0.461	0.403	0.388	0.463	0.416	0
13	BOL + split	0.459	0.397	0.377	0.455	0.413	5
15	BOL	0.445	0.428	0.432	0.461	0.420	4
15	BOL + split	0.470	0.429	0.440	0.457	0.430	1
17	BOL	0.456	0.422	0.429	0.457	0.438	4
17	BOL + split	0.458	0.433	0.435	0.448	0.446	1
19	BOL	0.439	0.437	0.449	0.443	0.459	3
19	BOL + split	0.455	0.441	0.446	0.436	0.466	2
5	BOL_{stop}	0.451	0.270	0.280	0.480	0.271	0
5	BOL_{stop} + split (d100)	0.439	0.264	0.274	0.479	0.267	5
7	BOL_{stop}	0.379	0.392	0.315	0.541	0.538	0.5
7	BOL_{stop} + split (d100)	0.379	0.391	0.314	0.482	0.504	4.5
9	BOL_{stop}	0.352	0.528	0.400	0.427	0.475	0
9	BOL_{stop} + split (d100)	0.341	0.513	0.396	0.419	0.450	5
11	BOL_{stop}	0.440	0.393	0.424	0.424	0.357	2
11	BOL_{stop} + split (d100)	0.439	0.374	0.410	0.429	0.365	3
13	BOL_{stop}	0.454	0.402	0.392	0.459	0.413	0
13	BOL_{stop} + split (d100)	0.449	0.396	0.384	0.455	0.410	5
15	BOL_{stop}	0.449	0.440	0.432	0.473	0.416	3
15	BOL_{stop} + split (d100)	0.451	0.435	0.434	0.455	0.420	2
17	BOL_{stop}	0.454	0.451	0.431	0.459	0.431	2
17	BOL_{stop} + split (d100)	0.456	0.433	0.432	0.448	0.427	3
19	BOL_{stop}	0.461	0.457	0.450	0.438	0.456	1.5
19	BOL_{stop} + split (d100)	0.461	0.436	0.457	0.429	0.453	3.5

Table D.26: **Subset experiments for compound splitting (AMAZON).** [Section 6.1.4]

subset size (labels)		subset 1	subset 2	subset 3	subset 4	subset 5	eval
5	BOL	0.288	0.386	0.246	0.304	0.372	3
5	BOL + split	0.297	0.346	0.248	0.340	0.358	2
7	BOL	0.193	0.325	0.422	0.500	0.195	2
7	BOL + split	0.187	0.324	0.423	0.398	0.203	3
9	BOL	0.327	0.272	0.390	0.337	0.430	0
9	BOL + split	0.309	0.252	0.380	0.320	0.418	5
11	BOL	0.386	0.375	0.370	0.373	0.400	2
11	BOL + split	0.408	0.366	0.390	0.369	0.376	3
13	BOL	0.386	0.388	0.364	0.406	0.379	1
13	BOL + split	0.378	0.400	0.347	0.402	0.367	4
15	BOL	0.389	0.387	0.384	0.384	0.375	3
15	BOL + split	0.378	0.405	0.399	0.391	0.372	2
17	BOL	0.404	0.438	0.381	0.390	0.372	3
17	BOL + split	0.406	0.420	0.388	0.408	0.367	2
19	BOL	0.397	0.416	0.389	0.430	0.414	2
19	BOL + split	0.391	0.404	0.394	0.403	0.418	3
5	BOL_{stop}	0.274	0.362	0.243	0.337	0.372	1
5	BOL_{stop} + split (d100)	0.250	0.340	0.238	0.364	0.360	4
7	BOL_{stop}	0.199	0.298	0.435	0.416	0.189	2
7	BOL_{stop} + split (d100)	0.181	0.313	0.424	0.394	0.190	3
9	BOL_{stop}	0.320	0.270	0.389	0.314	0.410	3
9	BOL_{stop} + split (d100)	0.326	0.286	0.348	0.313	0.462	2
11	BOL_{stop}	0.394	0.371	0.366	0.353	0.388	2
11	BOL_{stop} + split (d100)	0.400	0.351	0.352	0.369	0.353	3
13	BOL_{stop}	0.392	0.402	0.338	0.393	0.349	4
13	BOL_{stop} + split (d100)	0.384	0.406	0.360	0.395	0.365	1
15	BOL_{stop}	0.382	0.405	0.382	0.398	0.371	3
15	BOL_{stop} + split (d100)	0.377	0.401	0.444	0.479	0.581	2
17	BOL_{stop}	0.401	0.428	0.384	0.393	0.348	2
17	BOL_{stop} + split (d100)	0.381	0.411	0.392	0.390	0.351	3
19	BOL_{stop}	0.377	0.417	0.397	0.412	0.399	0
19	BOL_{stop} + split (d100)	0.365	0.408	0.391	0.407	0.392	5

Table D.27: **Subset experiments for compound splitting (WIKI).** [Section 6.1.4]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
Shared bigrams (ordered)	0.780[a](25%)	0.618[b](55%)	0.457 (58%)	0.499[c](41%)	0.456 (46%)
Combined 1:1	0.419[0.012]	0.481[0.013]	0.404[0.011]	0.402[0.011]	0.368[0.029]
Combined 1:2	0.477[0.011]	0.485[0.007]	0.403[0.010]	0.405[0.017]	0.392[0.029]
Combined 1:3	0.510[0.015]	0.519[0.018]	0.410[0.018]	0.430[0.016]	0.446[0.003]
Combined 1:10	0.613[0.014]	0.542[0.011]	0.414[0.010]	0.440[0.014]	0.447[0.005]
Features BOL_{stop}, shared	11,539	187,797	142,217	265,150	248,591
Features BOL_{stop}, total	35,087	440,883	317,783	757,072	618,225
Bigrams, shared	11,662	697,427	616,568	582,724	997,461
Bigrams, total	118,570	2,878,712	2,859,784	4,292,656	6,788,070

[a]28 documents lost.
[b]33 documents lost.
[c]112 documents lost.

Table D.28: **Clustering with *bigram* features.** "Combined 1:10" means that the frequencies for the bigrams were multiplied by an extra factor of ten, etc. (before tf-idf weighting). The percentages in parentheses refer, as before, to the number of non-zero elements, whereas the number in the last four rows refer to the number of dimensions. Therefore the two do not stand in a direct relationship. [Section 6.2.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
Multi-part names	0.616[a](0.08%)	0.692[b](1.03%)	0.776[c](1.22%)	0.763[d](1.71%)	0.795[e](0.79%)
Combined 1:1	0.414[0.017]	0.461[0.005]	0.420[0.000]	0.412[0.020]	0.345[0.002]
Combined 1:2	0.411[0.018]	0.470[0.009]	0.417[0.002]	0.412[0.017]	0.343[0.002]
Combined 1:3	0.422[0.017]	0.470[0.008]	0.417[0.004]	0.414[0.019]	0.342[0.002]
Combined 1:10	0.414[0.016]	0.484[0.008]	0.423[0.004]	0.429[0.013]	0.345[0.001]
Shared BOL_{stop} features	11,539	187,797	142,217	265,150	248,591
Total BOL_{stop} features	35,087	440,883	317,783	757,072	618,225
Shared multi-part names	37	10,862	6,883	17,369	11,150
Total multi-part names	441	44,027	22,002	67,337	44,145

[a]3721 documents lost.
[b]39661 documents lost.
[c]36013 documents lost.
[d]31935 documents lost.
[e]9424 documents lost.

Table D.29: **Clustering with *multi-part name* features.** [Section 6.2.2]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
Noun phrases (NPs)	0.900[a](5%)	0.729[b](9%)	0.546[c](9%)	0.650[d](6%)	0.467[e](8%)
Combined 1:1	0.426[0.012]	0.452[0.005]	0.411[0.001]	0.415[0.016]	0.343[0.004]
Combined 1:2	0.442[0.010]	0.466[0.011]	0.410[0.002]	0.411[0.016]	0.337[0.001]
Combined 1:3	0.457[0.018]	0.492[0.006]	0.409[0.014]	0.418[0.018]	0.344[0.010]
Combined 1:10	0.588[0.020]	0.529[0.004]	0.426[0.015]	0.407[0.013]	0.407[0.027]
Features BOL_{stop}, shared	11,539	187,797	142,217	265,150	248,591
Features BOL_{stop}, total	35,087	440,883	317,783	757,072	618,225
Noun phrases, shared	3,781	202,438	170,374	130,687	238,804
Noun phrases, total	55,421	948,115	1,072,672	1,427,018	2,549,167

[a] 424 documents lost.
[b] 6312 documents lost.
[c] 462 documents lost.
[d] 5750 documents lost.
[e] 264 documents lost.

Table D.30: **Clustering with *noun phrases*.** [Section 6.2.3]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop} + ...					
—	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
syn **as**	0.567 (163.53%)	0.519 (165.40%)	0.527 (171.96%)	0.421 (157.89%)	0.362 (171.02%)
syn **fs**	0.428 (99.70%)	0.463 (99.51%)	0.429 (99.95%)	0.408 (99.40%)	0.350 (99.48%)
syn **fs+cep**	0.425 (99.75%)	0.460 (99.54%)	0.428 (99.96%)	0.402 (99.43%)	0.350 (99.52%)
syn **fs+o**	0.435 (99.53%)	0.465 (99.45%)	0.428 (99.88%)	0.408 (99.23%)	0.348 (99.38%)
syn **fs+hy[1]**	0.419 (100.13%)	0.467 (99.66%)	0.461 (99.11%)	0.401 (97.71%)	0.353 (98.32%)
syn **fs+hy[2]**	0.389 (103.03%)	0.476 (102.41%)	0.461 (102.67%)	0.398 (99.29%)	0.347 (99.63%)
syn **fs+hy[3]**	0.456 (106.36%)	0.496 (103.53%)	0.455 (103.73%)	0.437 (98.06%)	0.341 (96.49%)
syn **fs+ho**	0.454 (109.80%)	0.487 (109.42%)	0.450 (112.81%)	0.414 (108.89%)	0.355 (108.44%)
syn **ds**	0.429 (99.42%)	0.461 (99.10%)	0.422 (99.14%)	0.393 (98.83%)	0.351 (98.66%)

Table D.31: **Clustering with synsets.** [Section 6.3.3]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
BOL_{stop} + syn ...					
fs	0.428 (-0.30%)	0.463 (-0.49%)	0.429 (-0.05%)	0.408 (-0.60%)	0.350 (-0.52%)
fs+A	0.422 (-0.05%)	0.460 (-0.08%)	0.422 (-0.03%)	0.406 (-0.07%)	0.345 (-0.10%)
fs+N	0.421 (-0.23%)	0.465 (-0.37%)	0.425 (+0.02%)	0.394 (-0.48%)	0.351 (-0.37%)
fs+V	0.423 (-0.02%)	0.463 (-0.03%)	0.423 (-0.04%)	0.404 (-0.04%)	0.348 (-0.06%)
fs+a1	0.434 (-0.18%)	0.462 (-0.28%)	0.420 (-0.28%)	0.401 (-0.35%)	0.353 (-0.33%)
fs+a10	0.408 (-0.30%)	0.460 (-0.48%)	0.427 (-0.05%)	0.415 (-0.60%)	0.351 (-0.52%)
fs+a2	0.423 (-0.24%)	0.460 (-0.39%)	0.422 (-0.31%)	0.407 (-0.50%)	0.353 (-0.47%)
fs+a3	0.426 (-0.31%)	0.459 (-0.46%)	0.428 (-0.37%)	0.415 (-0.59%)	0.350 (-0.56%)
fs+a4	0.430 (-0.30%)	0.463 (-0.46%)	0.428 (-0.08%)	0.407 (-0.60%)	0.349 (-0.51%)
fs+df01	0.416 (+0.00%)	0.464 (+0.00%)	0.422 (+0.00%)	0.406 (-0.01%)	0.348 (+0.00%)
fs+df05	0.421 (+0.00%)	0.465 (-0.02%)	0.421 (-0.02%)	0.403 (-0.04%)	0.347 (-0.01%)
fs+df1	0.417 (+0.00%)	0.462 (-0.04%)	0.421 (-0.03%)	0.402 (-0.06%)	0.347 (-0.01%)
fs+df2	0.410 (+0.00%)	0.462 (-0.07%)	0.420 (-0.06%)	0.404 (-0.10%)	0.346 (-0.02%)
fs+df5	0.416 (-0.01%)	0.463 (-0.15%)	0.422 (-0.09%)	0.402 (-0.18%)	0.349 (-0.07%)
fs+df10	0.431 (-0.02%)	0.462 (-0.18%)	0.421 (-0.14%)	0.411 (-0.28%)	0.350 (-0.12%)
fs+df25	0.427 (-0.08%)	0.462 (-0.27%)	0.425 (-0.25%)	0.418 (-0.43%)	0.348 (-0.21%)
fs+df50	0.425 (-0.14%)	0.463 (-0.34%)	0.429 (-0.25%)	0.411 (-0.50%)	0.351 (-0.34%)
fs+sf50	0.417 (-0.14%)	0.465 (-0.36%)	0.427 (-0.34%)	0.407 (-0.50%)	0.350 (-0.48%)
fs+sf250	0.430 (-0.12%)	0.464 (-0.31%)	0.425 (-0.29%)	0.408 (-0.40%)	0.354 (-0.36%)
fs+sf500	0.416 (-0.11%)	0.464 (-0.26%)	0.425 (-0.27%)	0.412 (-0.37%)	0.351 (-0.29%)
fs+si50	0.419 (-0.28%)	0.464 (-0.48%)	0.427 (-0.05%)	0.407 (-0.59%)	0.348 (-0.52%)
fs+si100	0.426 (-0.28%)	0.465 (-0.47%)	0.428 (-0.05%)	0.393 (-0.59%)	0.351 (-0.51%)
fs+si150	0.421 (-0.28%)	0.464 (-0.47%)	0.427 (-0.04%)	0.403 (-0.58%)	0.351 (-0.50%)
fs+si250	0.418 (-0.25%)	0.462 (-0.40%)	0.429 (-0.01%)	0.402 (-0.54%)	0.352 (-0.46%)
fs+si500	0.412 (-0.23%)	0.464 (-0.35%)	0.426 (+0.00%)	0.400 (-0.52%)	0.351 (-0.42%)
fs+su10	0.421 (-0.24%)	0.464 (-0.36%)	0.428 (-0.05%)	0.407 (-0.53%)	0.351 (-0.49%)
fs+su50	0.413 (-0.16%)	0.464 (-0.34%)	0.427 (-0.29%)	0.406 (-0.50%)	0.353 (-0.45%)
fs+su100	0.426 (-0.13%)	0.463 (-0.31%)	0.426 (-0.24%)	0.396 (-0.44%)	0.352 (-0.39%)
fs+su250	0.404 (-0.10%)	0.464 (-0.26%)	0.424 (-0.21%)	0.412 (-0.36%)	0.352 (-0.30%)
fs+t2	0.404 (-0.01%)	0.462 (-0.10%)	0.421 (-0.09%)	0.419 (-0.09%)	0.347 (-0.03%)
fs+t5	0.410 (-0.02%)	0.462 (-0.32%)	0.420 (-0.12%)	0.409 (-0.13%)	0.350 (-0.06%)
fs+t10	0.412 (-0.04%)	0.464 (-0.42%)	0.422 (-0.16%)	0.411 (-0.21%)	0.350 (-0.08%)
fs+t20	0.416 (-0.07%)	0.463 (-0.48%)	0.423 (-0.23%)	0.401 (-0.33%)	0.352 (-0.14%)
fs+t50	0.421 (-0.11%)	0.464 (-0.34%)	0.430 (-0.27%)	0.412 (-0.46%)	0.351 (-0.27%)
fs+cep	0.425 (-0.25%)	0.460 (-0.46%)	0.428 (-0.04%)	0.402 (-0.57%)	0.350 (-0.48%)
fs+cep+NA	0.431 (-0.23%)	0.466 (-0.42%)	0.428 (+0.01%)	0.398 (-0.53%)	0.352 (-0.43%)
fs+o	0.435 (-0.47%)	0.465 (-0.55%)	0.428 (-0.12%)	0.408 (-0.77%)	0.348 (-0.62%)
fs+o+N	0.438 (-0.33%)	0.465 (-0.37%)	0.423 (-0.01%)	0.410 (-0.53%)	0.349 (-0.38%)
fs+o+Ndf2	0.413 (+0.01%)	0.464 (-0.05%)	0.420 (-0.04%)	0.415 (-0.09%)	0.348 (-0.02%)
fs+o+Ndf5	0.407 (+0.00%)	0.464 (-0.11%)	0.421 (-0.07%)	0.404 (-0.16%)	0.346 (-0.05%)

continued on next page

continued from previous page

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop}	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
BOL_{stop} + syn ...					
fs+hy[1]	0.419 (+0.13%)	0.467 (-0.34%)	0.461 (-0.89%)	0.401 (-2.29%)	0.353 (-1.68%)
fs+hy[1]+A	0.429 (+0.03%)	0.462 (-0.19%)	0.425 (-0.03%)	0.398 (-0.25%)	0.353 (-0.39%)
fs+hy[1]+N	0.413 (+0.07%)	0.470 (-0.22%)	0.456 (-0.96%)	0.395 (-2.03%)	0.354 (-1.32%)
fs+hy[1]+m8	0.427 (-0.09%)	0.466 (-0.42%)	0.430 (-0.35%)	0.396 (-0.99%)	0.352 (-0.65%)
fs+hy[1]+m15	0.429 (+0.05%)	0.461 (-0.40%)	0.436 (-0.44%)	0.407 (-1.34%)	0.351 (-0.94%)
fs+hy[2]	0.389 (+3.03%)	0.476 (+2.41%)	0.461 (+2.67%)	0.398 (-0.71%)	0.347 (-0.37%)
fs+hy[2]+A	0.428 (+0.18%)	0.466 (-0.11%)	0.423 (+0.17%)	0.416 (-0.29%)	0.350 (-0.57%)
fs+hy[2]+N	0.429 (+2.55%)	0.477 (+1.82%)	0.460 (+1.92%)	0.395 (-0.63%)	0.348 (-0.10%)
fs+hy[2]+m8	0.415 (+0.31%)	0.466 (+0.66%)	0.422 (+0.52%)	0.394 (-0.26%)	0.349 (+0.10%)
fs+hy[2]+m15	0.412 (+0.91%)	0.468 (+1.06%)	0.431 (+1.39%)	0.411 (-0.30%)	0.346 (+0.35%)
fs+hy[3]	0.456 (+6.36%)	0.496 (+3.53%)	0.455 (+3.73%)	0.437 (-1.94%)	0.341 (-3.51%)
fs+hy[3]+A	0.431 (+0.29%)	0.469 (-0.66%)	0.420 (-0.17%)	0.404 (-1.05%)	0.344 (-2.40%)
fs+hy[3]+N	0.473 (+5.40%)	0.482 (+3.43%)	0.464 (+3.03%)	0.437 (-1.04%)	0.346 (-1.16%)
fs+hy[3]+m8	0.413 (+0.29%)	0.463 (+0.25%)	0.424 (+0.42%)	0.419 (+0.04%)	0.351 (+0.22%)
fs+hy[3]+m15	0.434 (+0.62%)	0.467 (+0.58%)	0.424 (+0.28%)	0.397 (-0.32%)	0.354 (+0.17%)
fs+ho	0.454 (+9.80%)	0.487 (+9.42%)	0.450 (+12.81%)	0.414 (+8.89%)	0.355 (+8.44%)
as	0.567 (+63.5%)	0.519 (+65.4%)	0.527 (+72.0%)	0.421 (+57.9%)	0.362 (+71.0%)
ds	0.429 (-0.58%)	0.461 (-0.90%)	0.422 (-0.86%)	0.393 (-1.17%)	0.351 (-1.34%)
ds+A	0.423 (-0.06%)	0.468 (-0.10%)	0.421 (-0.04%)	0.409 (-0.09%)	0.349 (-0.14%)
ds+V	0.420 (-0.05%)	0.462 (-0.08%)	0.423 (-0.11%)	0.402 (-0.13%)	0.349 (-0.18%)
ds+N	0.431 (-0.47%)	0.464 (-0.72%)	0.422 (-0.71%)	0.392 (-0.95%)	0.349 (-1.02%)
ds+Ndf2	0.410 (-0.03%)	0.464 (-0.09%)	0.422 (-0.08%)	0.407 (-0.13%)	0.349 (-0.03%)
ds+Ndf5	0.419 (-0.07%)	0.460 (-0.18%)	0.421 (-0.15%)	0.401 (-0.25%)	0.350 (-0.08%)
ds+Ndf10	0.424 (-0.11%)	0.463 (-0.28%)	0.421 (-0.24%)	0.399 (-0.38%)	0.351 (-0.14%)
ds+Ndf20	0.438 (-0.19%)	0.462 (-0.38%)	0.420 (-0.39%)	0.390 (-0.58%)	0.351 (-0.23%)
ds+a1	0.417 (-0.20%)	0.461 (-0.33%)	0.419 (-0.37%)	0.417 (-0.47%)	0.349 (-0.41%)
ds+a2	0.419 (-0.33%)	0.464 (-0.54%)	0.423 (-0.55%)	0.405 (-0.74%)	0.349 (-0.73%)
ds+a4	0.416 (-0.52%)	0.462 (-0.76%)	0.425 (-0.75%)	0.402 (-1.04%)	0.352 (-1.13%)

Table D.32: **Clustering with refined synset selection.** [Section 6.3.4]

D.4 Experiments in Chapter 7 (Combining)

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL + ...					
—	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
individual:					
prune	0.412 (60%)	0.451 (80%)	0.381 (32%)	0.382 (52%)	0.347 (60%)
stop	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)
pos1 (SUB, ADJ)	0.403 (54%)	0.457 (47%)	0.394 (46%)	0.405 (50%)	0.355 (54%)
pos2 (SUB, ADJ, NAM$_{all}$)	0.402 (57%)	0.468 (55%)	0.416 (53%)	0.393 (61%)	0.348 (59%)
wgt1 (SUB, ADJ)	0.415 (100%)	0.446 (100%)	0.446 (100%)	0.387 (100%)	0.344 (100%)
wgt2 (SUB, ADJ, NAM$_{all}$)	0.408 (100%)	0.455 (100%)	0.439 (100%)	0.388 (100%)	0.342 (100%)
combined:					
prune + stop	0.411 (46%)	0.459 (53%)	0.388 (31%)	0.406 (40%)	0.346 (54%)
pos1 + prune	0.403 (42%)	0.460 (43%)	0.384 (22%)	0.401 (34%)	0.364 (41%)
pos1 + stop	0.416 (50%)	0.461 (42%)	0.390 (43%)	0.401 (45%)	0.356 (49%)
pos1 + stop + prune	0.414 (38%)	0.457 (39%)	0.379 (22%)	0.398 (32%)	0.364 (39%)
pos2 + prune	0.397 (43%)	0.474 (50%)	0.403 (27%)	0.403 (38%)	0.343 (46%)
pos2 + stop	0.404 (52%)	0.465 (49%)	0.412 (49%)	0.409 (57%)	0.357 (54%)
pos2 + stop + prune	0.405 (39%)	0.463 (45%)	0.392 (26%)	0.406 (36%)	0.342 (44%)
wgt1 + stop	0.416 (61%)	0.465 (60%)	0.417 (59%)	0.398 (67%)	0.344 (67%)
wgt1 + prune	0.418 (60%)	0.449 (80%)	0.379 (32%)	0.390 (52%)	0.348 (60%)
wgt1 + stop + prune	0.418 (46%)	0.468 (53%)	0.379 (31%)	0.395 (40%)	0.348 (54%)
wgt2 + stop	0.418 (61%)	0.466 (60%)	0.416 (59%)	0.416 (67%)	0.343 (67%)
wgt2 + prune	0.409 (60%)	0.462 (80%)	0.390 (32%)	0.389 (52%)	0.344 (60%)
wgt2 + stop + prune	0.414 (46%)	0.462 (53%)	0.385 (31%)	0.403 (40%)	0.344 (54%)

Table D.33: **Combining matrix reduction techniques.** [Section 7.1]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
BOL_{stop} + ...					
—	0.422 (100%)	0.463 (100%)	0.421 (100%)	0.408 (100%)	0.349 (100%)
individual:					
split	0.360 (138%)	0.450 (122%)	0.373 (131%)	0.384 (124%)	0.346 (124%)
names	0.411 (100%)	0.470 (101%)	0.417 (101%)	0.412 (102%)	0.343 (101%)
NPs	0.442 (108%)	0.466 (115%)	0.410 (116%)	0.411 (109%)	0.337 (112%)
combined:					
split + names	0.357 (139%)	0.460 (123%)	0.392 (132%)	0.395 (126%)	0.342 (125%)
split + NPs	0.360 (147%)	0.465 (137%)	0.383 (146%)	0.397 (134%)	0.359 (136%)
names + NPs	0.437 (108%)	0.483 (116%)	0.400 (117%)	0.419 (111%)	0.336 (113%)
split + names + NPs	0.362 (147%)	0.484 (138%)	0.384 (147%)	0.387 (135%)	0.342 (137%)

Table D.34: **Combining matrix enhancement techniques.** [Section 7.2]

	SPRINGER	AMAZON	SDA	WIKI	NZZ
individual:					
BOL	0.421 (100%)	0.448 (100%)	0.456 (100%)	0.388 (100%)	0.349 (100%)
BOL + prune	0.412 (60%)	0.451 (80%)	0.381 (32%)	0.382 (52%)	0.347 (60%)
BOL + stop	0.422 (61%)	0.463 (60%)	0.421 (59%)	0.408 (67%)	0.349 (67%)
BOL + wgt1	0.415 (100%)	0.446 (100%)	0.446 (100%)	0.387 (100%)	0.344 (100%)
BOL + wgt2	0.408 (100%)	0.455 (100%)	0.439 (100%)	0.388 (100%)	0.342 (100%)
BOL + split	0.363 (124%)	0.453 (114%)	0.409 (119%)	0.390 (117%)	0.346 (117%)
BOL + NPs (+stop)	0.442 (65%)	0.466 (69%)	0.410 (69%)	0.411 (73%)	0.337 (75%)
combined:					
BOL + ...					
split + prune	0.372 (76%)	0.458 (88%)	0.391 (35%)	0.404 (55%)	0.347 (67%)
split + prune + NPs	0.388 (81%)	0.472 (97%)	0.397 (44%)	0.414 (61%)	0.367 (75%)
split + wgt1	0.369 (124%)	0.462 (114%)	0.398 (119%)	0.391 (117%)	0.355 (117%)
split + wgt1 + prune	0.371 (76%)	0.469 (88%)	0.389 (35%)	0.386 (55%)	0.360 (67%)
split + wgt1 + prune + NPs	0.368 (81%)	0.462 (97%)	0.385 (44%)	0.394 (61%)	0.359 (75%)
split + wgt2	0.360 (124%)	0.457 (114%)	0.394 (119%)	0.397 (117%)	0.347 (117%)
split + wgt2 + prune	0.367 (76%)	0.467 (88%)	0.396 (35%)	0.395 (55%)	0.351 (67%)
split + wgt2 + prune + NPs	0.363 (81%)	0.464 (97%)	0.396 (44%)	0.389 (61%)	0.351 (75%)
BOL_{stop} + ...					
split	0.360 (84%)	0.450 (73%)	0.373 (78%)	0.384 (83%)	0.346 (83%)
split + prune	0.363 (63%)	0.461 (63%)	0.401 (34%)	0.400 (46%)	0.348 (63%)
split + prune + NPs	0.374 (68%)	0.476 (73%)	0.398 (43%)	0.387 (52%)	0.368 (71%)
split + wgt1	0.360 (84%)	0.451 (73%)	0.375 (78%)	0.390 (83%)	0.355 (83%)
split + wgt1 + prune	0.358 (63%)	0.454 (63%)	0.392 (34%)	0.388 (46%)	0.360 (63%)
split + wgt1 + prune + NPs	0.360 (68%)	0.457 (73%)	0.387 (43%)	0.389 (52%)	0.359 (71%)
split + wgt2	0.363 (84%)	0.450 (73%)	0.379 (78%)	0.391 (83%)	0.348 (83%)
split + wgt2 + prune	0.366 (63%)	0.467 (63%)	0.393 (34%)	0.388 (46%)	0.353 (63%)
split + wgt2 + prune + NPs	0.363 (68%)	0.468 (73%)	0.390 (43%)	0.388 (52%)	0.353 (71%)

Table D.35: **Combining matrix enhancement and reduction techniques.** [Section 7.3]

Bibliography

[Agrawal *et al.*, 1993] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Database Mining: A Performance Perspective. *IEEE Transactions on Knowledge and Data Engineering*, **5**(6):914–925, 1993.

[Allan *et al.*, 1997] James Allan, James P. Callan, W. Bruce Croft, Lisa Ballesteros, Donald Byrd, Russell C. Swan, and Jinxi Xu. INQUERY Does Battle With TREC-6. In *Proceedings of the Sixth Text REtrieval Conference (TREC-6)*, pages 169–206, Gaithersburg, MD, 1997.

[Allen, 1992] Dennis Allen. Managing Infoglut. *BYTE*, **17**(6), 1992.

[Anderson *et al.*, 1999] Ed Anderson, Zhaojun Bai, Christian H. Bischof, Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, and Danny C. Sorensen. *LAPACK User's Guide, Third Edition*. SIAM, Philadelphia, 1999.
URL (last visited on 17 January 2006): `www.netlib.org/lapack`.

[Arampatzis *et al.*, 2000a] Avi Arampatzis, Theo P. van der Weide, Cornelis H.A. Koster, and Patrick van Bommel. An Evaluation of Linguistically-motivated Indexing Schemes. In *Proceedings of the 22nd BCS-IRSG Annual Colloquium on Information Retrieval Research*, pages 34–45, Cambridge, United Kingdom, 2000.

[Arampatzis *et al.*, 2000b] Avi Arampatzis, Theo P. van der Weide, Patrick van Bommel, and Cornelis H.A. Koster. Linguistically Motivated Information Retrieval. In *Encyclopedia of Library and Information Science*, volume 69, pages 201–222. Marcel Dekker, New York/Basle, 2000.

[Baeza-Yates and Ribeiro-Neto, 1999] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto, editors. *Modern Information Retrieval*. Addison-Wesley, Reading, MA, 1999.

[Baeza-Yates *et al.*, 2003] Ricardo A. Baeza-Yates, Benjamin Bustos, Edgar Chávez, Norma Herrera, and Gonzalo Navarro. Clustering in Metric Spaces with Applications to Information Retrieval. In Wu *et al.* (2003), pages 1–34.

[Bakus *et al.*, 2002] Jan Bakus, Mahmoud Hussin, and Mohamed S. Kamel. SOM-Based Document Clustering Using Phrases. In *Proceedings of the 9th International Conference on Neural Information Processing (ICONIP'2002)*, volume 5, pages 2212–2216, Singapore, 2002.

[Banfield and Raftery, 1993] Jeffrey D. Banfield and Adrian E. Raftery. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, **49**:803–821, 1993.

[Basili *et al.*, 2000] Roberto Basili, Alessandro Moschitti, and Maria Teresa Pazienza. Language Sensitive Text Classification. In *Proceedings of 6th International Conference 'Recherche d'Information Assistée par Ordinateur' (RIAO'00)*, pages 331–343, Paris, France, 2000.

[Bates, 1989] Marcia J. Bates. The Design of Browsing and Berrypicking Techniques for the Online Search Interface. *Online Review*, **13**(5):407–424, 1989.

[Beil *et al.*, 2002] Florian Beil, Martin Ester, and Xiaowei Xu. Frequent Term-Based Text Clustering. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 436–442, Edmonton, Canada, 2002.

[Bellot and El-Bèze, 2000] Patrice Bellot and Marc El-Bèze. Clustering by Means of Unsupervised Decision Trees or Hierarchical and K-means-like Algorithm. In *Proceedings of 6th International Conference 'Recherche d'Information Assistée par Ordinateur' (RIAO'00)*, pages 344–363, Paris, France, 2000.

[Berry and Linoff, 1997] Michael Berry and Gordon Linoff. *Data Mining Techniques: For Marketing, Sales and Customer Support.* John Wiley and Sons, New York, 1997.

[Berry *et al.*, 1993] Michael Berry, Theresa Do, Gavin O'Brien, Vijay Krishna, and Sowmini Varadhan. SVDPACKC (Version 1.0) User's Guide. Technical Report CS-93-194. Department of Computer Science, University of Tennessee, 1993. Revised March 1996.

[Boley *et al.*, 1999a] Daniel Boley, Maria Gini, Robert Gross, Eui-Hong (Sam) Han, Kyle Hastings, George Karypis, Vipin Kumar, Bamshad Mobasher, and Jerome Moore. Document Categorization and Query Generation on the World Wide Web Using WebACE. *Journal of Artificial Intelligence Review*, **13**(5–6):365–391, 1999.

[Boley *et al.*, 1999b] Daniel Boley, Maria Gini, Robert Gross, Eui-Hong (Sam) Han, Kyle Hastings, George Karypis, Vipin Kumar, Bamshad Mobasher, and Jerome Moore. Partitioning-Based Clustering for Web Document Categorization. *Decision Support Systems*, **27**(3):329–341, 1999.

[Boulis and Ostendorf, 2004] Constantinos Boulis and Mari Ostendorf. Combining Multiple Clustering Systems. In *8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2004)*, page 63, Pisa, Italy, 2004.

[Bowman *et al.*, 1994] C. Mic Bowman, Peter B. Danzig, Udi Manber, and Michael F. Schwartz. Scalable Internet Resource Discovery: Research Problems and Approaches. *Communications of the ACM*, **37**(8):98–107, 1994.

[Bradley and Fayyad, 1998] Paul S. Bradley and Usama M. Fayyad. Refining Initial Points for K-Means Clustering. In *Proceedings of the 15th International Conference on Machine Learning (ICML'98)*, pages 91–99, San Francisco, CA, 1998.

[Bradley *et al.*, 1998] Paul S. Bradley, Usama M. Fayyad, and Cory Reina. Scaling EM (Expectation-Maximization) Clustering to Large Databases. Technical Report MSR-TR-98-35. Microsoft Research, 1998. Revised 1999.

[Broder *et al.*, 1997] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic Clustering of the Web. In *Proceedings of the 6th International World Wide Web Conference (WWW-6)*, pages 391–404, Santa Clara, CA, 1997.

[Broder, 2002] Andrei Broder. A Taxonomy of Web Search. *SIGIR Forum*, **36**(2):3–10, 2002.

[Budzik and Hammond, 1999] Jan Budzik and Kristian Hammond. Watson: Anticipating and Contextualizing Information Needs. In *Proceedings of the 62nd Annual Meeting of the American Society for Information Science*, pages 727–740, Medford, NJ, 1999.

[Caropreso *et al.*, 2001] Maria Fernanda Caropreso, Stan Matwin, and Fabrizio Sebastiani. A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization. In A.G. Chin, editor, *Text Databases and Document Management: Theory and Practice*, pages 78–102. Idea Group Publishing, Hershey, 2001.

[Casillas *et al.*, 2003] Arantza Casillas, Mayte González de Lena, and Raquel Martínez. Partitional Clustering Experiments with News Documents. In *4th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing-03)*, pages 615–618, Mexico City, Mexico, 2003.

[Chang and Hsu, 1998] Chia-Hui Chang and Ching-Chi Hsu. Hypertext Information Retrieval for Short Queries. In *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop*, Taipei, Taiwan, 1998.

[Cheeseman and Stutz, 1996] Peter Cheeseman and John Stutz. Bayesian Classification (AutoClass): Theory and Results. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 61–83. AAAI Press/MIT Press, 1996.

[Cheeseman *et al.*, 1988] Peter Cheeseman, James Kelly, Matthew Self, John Stutz, Will Taylor, and Don Freeman. AutoClass: A Bayesian Classification System. In *Proceedings of the Fifth International Conference on Machine Learning (ICML'88)*, pages 54–64, Ann Arbor, MI, 1988.

[Choo *et al.*, 2000a] Chun Wei Choo, Brian Detlor, and Don Turnbull. Information Seeking on the Web: An Integrated Model of Browsing and Searching. *First Monday*, 2000.
URL (last visited on 17 January 2006): `http://firstmonday.org/issues/issue5_2/choo`.

[Choo *et al.*, 2000b] Chun Wei Choo, Brian Detlor, and Don Turnbull. *Web Work: Information Seeking and Knowledge Work on the World Wide Web*. Kluwer Academic Publishers, Dordrecht, 2000.

[Choudhary and Bhattacharyya, 2002] Bhoopesh Choudhary and Pushpak Bhattacharyya. Text Clustering using Semantics. In *Proceedings of the 11th International World Wide Web Conference (WWW-11)*, Honolulu, HI, 2002. (Poster).

[Chu *et al.*, 2002] Shu-Chuan Chu, John F. Roddick, and Jeng-Shyang Pan. An Incremental Multi-Centroid, Multi-Run Sampling Scheme for k-medoids-based Algorithms—Extended Report. Technical Report KDM-02-003. Flinders University, Adelaide, Australia, 2002.

[Chu *et al.*, 2003] Wesley W. Chu, Zhenyu Liu, and Wenlei Mao. Techniques for Textual Document Indexing and Retrieval Knowledge Sources and Data Mining. In Wu *et al.* (2003), pages 135–160.

[Clematide, 2002] Simon Clematide. Selektive Evaluation von robusten Parsern. In *Sechste Konferenz zur Verarbeitung natürlicher Sprache (Konvens 2002)*, pages 23–29, Saarbrücken, Germany, 2002.

[Clifton *et al.*, 2004] Chris Clifton, Robert Cooley, and Jason Rennie. TopCat: Data Mining for Topic Identification in a Text Corpus. *Transactions on Knowledge and Data Engineering*, **16**(8):949–964, 2004.

[Craswell and Hawking, 2002] Nick Craswell and David Hawking. Overview of the TREC-2002 Web Track. In *Proceedings of the 11th Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, 2002.

[Cristofor and Simovici, 2001] Dana Cristofor and Dan A. Simovici. An Information-Theoretical Approach to Genetic Algorithms for Clustering. Technical Report TR-01-02. University of Massachusetts, Boston, 2001.

[Croft *et al.*, 1991] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of the 14th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'91)*, pages 32–45, Chicago, IL, 1991.

[Cutting *et al.*, 1992] Douglass R. Cutting, David Karger, Jan Pedersen, and John W. Tukey. Scatter/Gather: A Cluster-Based Approach to Browsing Large Document Collections. In *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'92)*, pages 318–329, Copenhagen, Denmark, 1992.

[Cutting *et al.*, 1993] Douglass R. Cutting, David Karger, and Jan Pedersen. Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections. In *Proceedings of the 16th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 126–134, Pittsburgh, PA, 1993.

[Davies and Bouldin, 1979] David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **1**:224–227, 1979.

[Deerwester *et al.*, 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, **41**(6):391–407, 1990.

[Dempster *et al.*, 1977] Arthur Pentland Dempster, Nan M. Laird, and Donald B. Rubin. Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm (with Discussion). *Journal of the Royal Statistical Society, Series B*, **39**(1):1–38, 1977.

[Devlin and Burke, 1997] Brendan Devlin and Mary Burke. Internet: The Ultimate Reference Tool? *Internet Research: Electronic Networking Applications and Policy*, **7**(2):101–108, 1997.

[Dhillon and Modha, 2001] Inderjit S. Dhillon and Dharmendra S. Modha. Concept Decompositions for Large Sparse Text Data using Clustering. *Machine Learning*, **42**(1-2):143–175, 2001.

[Dhillon *et al.*, 2001] Inderijt S. Dhillon, James Fan, and Yuqiang Guan. Efficient Clustering of Very Large Document Collections. In R.L. Grossman, Ch. Kamath, V. Kumar, Ph. Kegelmeyer, and R.R. Naburu, editors, *Data Mining for Scientific and Engineering Applications*, pages 357–382. Kluwer Academic Publishers, Dordrecht, 2001.

[Dhillon *et al.*, 2002] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. Enhanced Word Clustering for Hierarchical Text Classification. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 191–120, Edmonton, Canada, 2002.

[Dhillon, 2001] Inderjit S. Dhillon. Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD-01)*, pages 269–274, San Francisco, CA, 2001. A longer version appears as Technical Report #2001-05, University of Texas, Austin, 2001.

[Ding and He, 2002] Chris H.Q. Ding and Xiaofeng He. Cluster Merging and Splitting in Hierarchical Clustering Algorithms. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 139–146, Maebashi City, Japan, 2002.

[Ding *et al.*, 2001] Chris H.Q. Ding, Xiaofeng He, Hongyuan Zha, Ming Gu, and Horst D. Simon. A Min-max Cut Algorithm for Graph Partitioning and Data Clustering. In *Proceedings of the First IEEE International Conference on Data Mining (ICDM 2001)*, pages 107–114, San Jose, CA, 2001.

[Dom, 2002] Byron E. Dom. An Information-Theoretic External Cluster-Validity Measure. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence (UAI '02)*, pages 137–145, Edmonton, Canada, 2002. Appears also as IBM Technical Report RJ 10219, 2001.

[Duda and Hart, 1973] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[El-Hamdouchi and Willett, 1986] Abdelmoula El-Hamdouchi and Peter Willett. Hierarchic Document Clustering Using Ward's Method. In *Proceedings of the 9th International ACM Conference on Research and Development in Information Retrieval (SIGIR'86)*, pages 149–156, Pisa, Italy, 1986.

[Ellis and Haugan, 1997] David Ellis and Merete Haugan. Modelling the Information Seeking Patterns of Engineers and Research Scientists in an Industrial Environment. *Journal of Documentation*, **53**(4):384–403, 1997.

[Ellis, 1989] David Ellis. A Behavioral Approach to Information Retrieval System Design. *Journal of Documentation*, **45**(3):171–212, 1989.

[Ertöz *et al.*, 2003] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding Topics in Collections of Documents: A Shared Nearest Neighbor Approach. In Wu *et al.* (2003), pages 83–104.

[Ester *et al.*, 1996] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2th International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, Portland, OR, 1996.

[Everitt, 1993] Brian S. Everitt. *Cluster Analysis*. Edward Arnold, London, 3rd edition, 1993.

[Fagan, 1989] Joel L. Fagan. The Effectiveness of a Nonsyntactic Approach to Automatic Phrase Indexing for Document Retrieval. *Journal of the American Society for Information Science*, **40**(2):115–132, 1989.

[Fasulo, 1999] Daniel Fasulo. An Analysis of Recent Work on Clustering Algorithms. Technical Report #01-03-02. University of Washington, 1999.

[Feldman, 1999] Susan Feldman. NLP Meets the Jabberwocky: Natural Language Processing in Information Retrieval. Online, **23**:62–72, 1999.

[Forgy, 1965] Edward W. Forgy. Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. *Biometrics*, **21**:768–769, 1965.

[Fox, 1992] Christopher Fox. Lexical Analysis and Stoplists. In Frakes and Baeza-Yates (1992), pages 102–130.

[Frakes and Baeza-Yates, 1992] William B. Frakes and Ricardo A. Baeza-Yates, editors. *Information retrieval: Data structures & Algorithms*. Prentice-Hall, Upper Saddle River, 1992.

[Frigui and Nasraoui, 2004] Hichem Frigui and Olfa Nasraoui. Simultaneous Clustering and Dynamic Keyword Weighting for Text Documents. In M.W. Berry, editor, *Survey of Text Mining*, pages 45–72. Springer, New York, 2004.

[Fung, 2002] Benjamin Chin Ming Fung. Hierarchical Document Clustering Using Frequent Itemsets. Master's thesis. Simon Fraser University, Burnaby, Canada, 2002.

[Goharian *et al.*, 2001] Nazli Goharian, Tarek El-Ghazawi, and David Grossman. Enterprise Text Processing: A Sparse Matrix Approach. In *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC-2001)*, pages 71–75, Las Vegas, NV, 2001.

[Goldszmidt and Sahami, 1998] Moises Goldszmidt and Mehran Sahami. A Probabilistic Approach to Full-Text Document Clustering. Technical Report ITAD-433-MS-98-044. SRI International, Menlo Park, CA, 1998.

[Golub and van Loan, 1996] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 3rd edition, 1996.

[Gonzalo *et al.*, 1998] Julio Gonzalo, Felisa Verdejo, Irina Chugur, and Juan Cigarran. Indexing with WordNet Synsets Can Improve Text Retrieval. In *Proceedings of the COLING/ACL '98 Workshop on Usage of WordNet for NLP*, pages 38–44, Montreal, Canada, 1998.

[Govindarajan and Ward, 1999] Jayesh Govindarajan and Matthew O. Ward. GeoViser—Geographic Visualization of Search Engine Results. In *10th International Workshop on Database and Expert Systems Applications (DEXA-1999)*, pages 269–273, Florence, Italy, 1999.

[Guha *et al.*, 1998] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an Efficient Clustering Algorithm for Large Databases. In *Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD-98)*, pages 73–84, Seattle, WA, 1998.

[Guha *et al.*, 2003] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Techniques for Clustering Massive Data Sets. In Wu *et al.* (2003), pages 35–82.

[Gulli and Signorini, 2005] Antonio Gulli and Alessio Signorini. The Indexable Web is More than 11.5 Billion Pages. In *Proceedings of the 14th International World Wide Web Conference*, pages 902–903, Chiba, Japan, 2005.

[Haapalainen and Majorin, 1995] Mariikka Haapalainen and Ari Majorin. GERTWOL und Morphologische Disambiguierung für das Deutsche. In *Proceedings of the 10th Nordic Conference of Computational Linguistics (NoDaLiDa-95)*, Helsinki, Finland, 1995.

[Halkidi *et al.*, 2000] Maria Halkidi, Michalis Vazirgiannis, and Yannis Batistakis. Quality Scheme Assessment in the Clustering Process. In *4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000)*, pages 265–276, Lyon, France, 2000.

[Halkidi *et al.*, 2001] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, **17**(2–3):107–145, 2001.

[Halkidi *et al.*, 2002] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. Cluster Validity Methods. *SIGMOD Record*, **31**(2, 3):40–45, 19–27, 2002.

[Hammouda and Kamel, 2002] Khaled M. Hammouda and Mohamed S. Kamel. Phrase-based Document Similarity Based on an Index Graph Model. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 203–210, Maebashi City, Japan, 2002.

[Hammouda, 2001] Khaled M. Hammouda. Web Mining: Clustering Web Documents, A Preliminary Review. Technical report. University of Waterloo, Ontario, Canada, 2001.

[Hamp and Feldweg, 1997] Birgit Hamp and Helmut Feldweg. GermaNet—a Lexical-Semantic Net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain, 1997.

[Hannappel *et al.*, 1999] Peter Hannappel, Reinhold Klapsing, Gustaf Neumann, and Adrian Krug. MSEEC—A Multi Search Engine with Multiple Clustering. In *Proceedings of the 10th Information Resources Management Association International Conference (IRMA'99)*, Hershey, PA, 1999.

[Harper *et al.*, 1999] David J. Harper, Mourad Mechkour, and Gheorghe Muresan. Document Clustering for Mediated Information Access. In *Proceedings of the 21st BCS-IRSG Annual Colloquium on IR Research*, pages 92–107, Glasgow, Scotland, 1999.

[Hasan and Matsumoto, 1999] Md Maruf Hasan and Yuji Matsumoto. Document Clustering: Before and After the Singular Value Decomposition. Technical Report TR:99. Information Processing Society of Japan, Sapporo, Japan, 1999. (SIGNotes Natural Language 134, pages 47–55).

[Hatzivassiloglou *et al.*, 2000] Vasileios Hatzivassiloglou, Luis Gravano, and Ankineedu Maganti. An Investigation of Linguistic Features and Clustering Algorithms for Topical Document Clustering. In *Proceedings of the 23rd International ACM Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 224–231, Athens, Greece, 2000.

[He *et al.*, 2001] Xiaofeng He, Chris H.Q. Ding, Hongyuan Zha, and Horst D. Simon. Automatic Topic Identification Using Webpage Clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'01)*, pages 195–202, San Jose, CA, 2001.

[He *et al.*, 2002] Ji He, Ah-Hwee Tan, and Chew-Lim Tan. ART-C: a Neural Architecture for Self-Organization under Constraints. In *Proceedings of the 2002 International Joint Conference on Neural Networks (IJCNN '02)*, volume 3, pages 2550–2555, Honolulu, HI, 2002.

[He *et al.*, 2003] Ji He, Ah-Hwee Tan, Chew Lim Tan, and Sam Yuan Sung. On Quantitative Evaluation of Clustering Systems. In Wu *et al.* (2003), pages 105–134.

[Hearst and Pedersen, 1996] Marti A. Hearst and Jan O. Pedersen. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *Proceedings of the 19th International ACM Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 76–84, Zurich, Switzerland, 1996.

[Hearst, 1999a] Marti A. Hearst. Untangling Text Data Mining. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics (ACL'99)*, pages 3–10, College Park, MD, 1999.

[Hearst, 1999b] Marti A. Hearst. The Use of Categories and Clusters for Organizing Retrieval Results. In Strzalkowski (1999), pages 333–374.

[Hearst, 1999c] Marti A. Hearst. User Interfaces and Visualization. In Baeza-Yates and Ribeiro-Neto (1999), pages 257–339.

[Henderson *et al.*, 2002a] James Henderson, Paola Merlo, Ivan Petroff, and Gerold Schneider. Using NLP to Efficiently Visualize Text Collections with SOMs. In *Proceedings of the 3rd International Workshop on Natural Language and Information Systems (NLIS 2002)*, pages 210–214, Aix-en-Provence, France, 2002.

[Henderson *et al.*, 2002b] James Henderson, Paola Merlo, Ivan Petroff, and Gerold Schneider. Using Syntactic Analysis to Increase Efficiency in Visualizing Text Collections. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 335–341, Taipei, Taiwan, 2002.

[Henzinger *et al.*, 2002] Monika R. Henzinger, Rajeev Motwani, and Craig Silverstein. Challenges in Web Search Engines. *SIGIR Forum*, **36**(2):11–22, 2002.

[Henzinger *et al.*, 2003] Monika R. Henzinger, Bay-Wei Chang, Brian Milch, and Sergey Brin. Query-free News Search. In *Proceedings of the 12th International World Wide Web Conference (WWW-12)*, pages 1–10, Budapest, Hungary, 2003.

[Hettich and Bay, 1999] Seth Hettich and Steven D. Bay. The UCI KDD Archive. 1999. URL (last visited on 17 January 2006): `http://kdd.ics.uci.edu`.

[Hofmann, 1999a] Thomas Hofmann. The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, volume 2, pages 682–687, Stockholm, Sweden, 1999.

[Hofmann, 1999b] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 50–57, Berkeley, CA, 1999.

[Hotho *et al.*, 2002] Andreas Hotho, Alexander Maedche, and Steffen Staab. Text Clustering Based on Good Aggregations. *Künstliche Intelligenz (KI)*, **16**(4):48–54, 2002.

[Hull, 1993] David Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proceedings of the 16th International ACM Conference on Research and Development in Information Retrieval (SIGIR'93)*, pages 329–338, Pittsburgh, PA, 1993.

[Jain and Dubes, 1988] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, 1988.

[Jancey, 1966] R.C. Jancey. Multidimensional Group Analysis. *Australian Journal of Botany*, **14**:127–130, 1966.

[Jardine and van Rijsbergen, 1971] Nicholas Jardine and Cornelis J. van Rijsbergen. The Use of Hierarchic Clustering in Information Retrieval. *Information Storage and Retrieval*, **7** (5):217–240, 1971.

[Jones *et al.*, 1995] Gareth Jones, Alexander M. Robertson, Chawchat Santimetvirul, and Peter Willett. Non-Hierarchic Document Clustering Using a Genetic Algorithm. *Information Research News*, **5**(3):2–6, 1995.

[Joshi and Jiang, 2001] Anupam Joshi and Zhihua Jiang. Retriever: Improving Web Search Engine Results Using Clustering. In A. Gangopadhyay, editor, *Managing Business With Electronic Commerce: Issues and Trends*, pages 59–81. Idea Group Publishing, Hershey, 2001.

[Jurafsky and Martin, 2000] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, 2000.

[Kanejiya *et al.*, 2004] Dharmendra Kanejiya, Arun Kumar, and Surendra Prasad. Statistical Language Modeling with Performance Benchmarks using Various Levels of Syntactic-Semantic Information. In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, pages 1161–1167, Geneva, Switzerland, 2004.

[Kanerva *et al.*, 2000] Pentti Kanerva, Jan Kristofersson, and Anders Holst. Random Indexing of Text Samples for Latent Semantic Analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, page 1036, Philadelphia, PA, 2000.

[Karlgren, 1999] Jussi Karlgren. Stylistic Experiments in Information Retrieval. In Strzalkowski (1999), pages 147–166.

[Karypis *et al.*, 1999] George Karypis, Eui-Hong (Sam) Han, and Vipin Kumar. Chameleon: A Hierarchical Clustering Algorithm Using Dynamic Modeling. *IEEE Computer*, **32**(8):68–75, 1999.

[Karypis, 2003] George Karypis. Cluto—A Clustering Toolkit. Technical Report RE #02-017. University of Minnesota, Department of Computer Science, 2003. Release 2.1.1.

[Kaski *et al.*, 1998] Samuel Kaski, Timo Honkela, Krista Lagus, and Teuvo Kohonen. WEBSOM—Self-Organizing Maps of Document Collections. *Neurocomputing*, **21**(1–3):101–117, 1998.

[Kaufman and Rousseeuw, 1990] Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data*. John Wiley and Sons, New York, 1990.

[Kilgariff, 1997] Adam Kilgariff. Using Word Frequency Lists to Measure Corpus Homogeneity and Similarity between Corpora. In *Proceedings of the 5th ACL SIGDAT Workshop on Very Large Corpora*, pages 231–245, Beijing and Hong Kong, China, 1997.

[Kilgariff, 2001] Adam Kilgariff. Comparing Corpora. *International Journal of Corpus Linguistics*, **6**(1):1–37, 2001.

[Kleinberg, 1998] Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 668–677, January 1998. Extended version in the *Journal of the ACM*, **46**(5):604–632, 1999. Also appears as IBM Research Report RJ 10076, 1997.

[Kohonen, 1984] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer, Berlin, 1st edition, 1984.

[Kohonen, 2001] Teuvo Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.

[Korfhage, 1991] Robert Korfhage. To See, or Not to See: Is *That* the Query? In *Proceedings of the 14th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'91)*, pages 134–141, Chicago, IL, 1991.

[Kraaij and Pohlmann, 1998] Wessel Kraaij and Renée Pohlmann. Comparing the Effect of Syntactic vs. Statistical Phrase Indexing Strategies for Dutch. In *Proceedings of Second European Conference on Research and Advanced Technology for Digital Librarie (ECDL '98)*, pages 605–614, Heraklion, Crete, Greece, 1998.

[Kreulen *et al.*, 2001] Jeff Kreulen, Dharmendra Modha, W. Scott Spangler, and Ray Strong. An Interactive Approach to Document Classification. Technical report. IBM Almaden Research Center, 2001. US Patent 6,424,971.

[Krishnapuram *et al.*, 1999] Raghu Krishnapuram, Anupam Joshi, and Liyu Yi. A Fuzzy Relative of the k-Medoids Algorithm with Application to Web Document and Snippet Clustering. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZIEEE-99)*, pages 1281–1286, Seoul, South Korea, 1999.

[Lancaster, 1968] Frederick Wilfrid Lancaster. *Information Retrieval Systems: Characteristics, Testing and Evaluation*. John Wiley and Sons, New York, 1968.

[Lance and Williams, 1967a] Godfrey N. Lance and William Thomas Williams. A General Theory of Classificatory Sorting Strategies. I. Hierarchical Systems. *The Computer Journal*, **9**:373–380, 1967.

[Lance and Williams, 1967b] Godfrey N. Lance and William Thomas Williams. A General Theory of Classificatory Sorting Strategies. II. Clustering Systems. *The Computer Journal*, **10**:271–277, 1967.

[Lang, 1995] Ken Lang. Newsweeder: Learning to Filter Netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pages 331–339, Tahoe City, CA, 1995.

[Large *et al.*, 1999] J. Andrew Large, Lucy A. Tedd, and Richard J. Hartley. *Information Seeking in the Online Age: Principles and Practice*. Bowker-Saur, London, 1999.

[Larocca Neto *et al.*, 2000] Joel Larocca Neto, Alexandre D. Santos, Celso A.A. Kaestner, and Alex A. Freitas. Document Clustering and Text Summarization. In *Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000)*, pages 41–55, London, United Kingdom, 2000.

[Larsen and Aone, 1999] Bjorner Larsen and Chinatsu Aone. Fast and Effective Text Mining Using Linear-Time Document Clustering. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pages 16–22, San Diego, CA, 1999.

[Leacock *et al.*, 1998] Claudia Leacock, George A. Miller, and Martin Chodorow. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, **24**(1): 147–165, 1998.

[Lee *et al.*, 2005] Michael D. Lee, Brandon Pincombe, and Matthew Welsh. An Empirical Evaluation of Models of Text Document Similarity. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society (CogSci2005)*, pages 1254–1259, Austin, TX, 2005.

[Lerman, 1999] Kristina Lerman. Document Clustering in Reduced Dimension Vector Space. URL (last visited on 17 January 2006): `www.isi.edu/~lerman/papers/Lerman99.pdf`. January 1999, Unpublished.

[Leuski, 2001] Anton Leuski. Evaluating Document Clustering for Interactive Information Retrieval. In *Proceedings of the 10th International ACM Conference on Information and Knowledge Management (CIKM 2001)*, pages 33–40, Atlanta, GA, 2001.

[Lewis, 1996] David Lewis. Dying for Information: An Investigation Into the Effects of Information Overload in the USA and Worldwide. Technical report. Reuters Limited, London, United Kingdom, 1996.

[Lewis, 1997] David D. Lewis. Reuters-21578 Text Categorization Test Collection Distribution 1.0. 1997.
URL (last visited on 17 January 2006): `www.daviddlewis.com/resources/testcollections`.

[Li *et al.*, 2004] Tao Li, Sheng Ma, and Mitsunori Ogihara. Document Clustering via Adaptive Subspace Iteration. In *Proceedings of the 27th Annual International Conference on Research and Development in Information Retrieval (SIGIR'04)*, pages 218–225, Sheffield, United Kingdom, 2004.

[Liu *et al.*, 2002] Xin Liu, Yihong Gong, Wei Xu, and Shenghuo Zhu. Document Clustering with Cluster Refinement and Model Selection Capabilities. In *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 191–198, Tampere, Finland, 2002.

[Luhn, 1958] Hans Peter Luhn. The Automatic Creation of Literature Abstracts. *IBM Journal of Research and Development*, **2**(2):159–165, 1958.

[Maarek and Smadja, 1989] Yoëlle S. Maarek and Frank A. Smadja. Full Text Indexing Based on Lexical Relations. An Application: Software Libraries. In *Proceedings of the 12th International ACM Conference on Research and Development in Information Retrieval (SIGIR'89)*, pages 198–206, Cambridge, MA, 1989.

[Maarek *et al.*, 2000] Yoëlle S. Maarek, Ronald Fagin, Israel Z. Ben-Shaul, and Dan Pelleg. Ephemeral Document Clustering for Web Applications. Technical Report RJ 10186. IBM, 2000.

[MacQueen, 1967] James B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, CA, 1967.

[Macskassy *et al.*, 1998] Sofus A. Macskassy, Arunava Banerjee, Brian D. Davison, and Haym Hirsh. Human Performance on Clustering Web Pages: A Preliminary Study. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 264–268, New York, NY, 1998.

[Manning and Schütze, 1999] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[Marchionini and Shneiderman, 1988] Gary Marchionini and Ben Shneiderman. Finding Facts vs. Browsing Knowledge in Hypertext Systems. *IEEE Computer*, **21**(1):70–80, 1988.

[Marchionini, 1989] Gary Marchionini. Information Seeking in Electronic Encyclopedias. *Machine-Mediated Learning*, **3**:211–226, 1989.

[Marchionini, 1992] Gary Marchionini. Interfaces for End-User Information Seeking. *Journal of the American Society for Information Science*, **43**(2):156–163, 1992.

[Marchionini, 1995] Gary Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, Cambridge, MA, 1995.

[Matthiesen, 1999] Martin Matthiesen. Morphologie im Textmining. Master's thesis. Universität Bielefeld, 1999.

[Merlo *et al.*, 2003] Paola Merlo, James Henderson, Gerold Schneider, and Eric Wehrli. Learning Document Similarity Using Natural Language Processing. *Linguistik online*, **17**:99–115, 2003.

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Cristiane Fellbaum, Derek Gross, and Katherine J. Miller. WordNet: An On-line Lexical Database. *International Journal of Lexicography*, **3**(4):235–312, 1990.

[Mitra *et al.*, 1997] Mandar Mitra, Chris Buckley, Amit Singhal, and Claire Cardie. An Analysis of Statistical and Syntactic Phrases. In *Proceedings of 5th International Conference 'Recherche d'Information Assistée par Ordinateur' (RIAO'97)*, pages 200–214, Montreal, Canada, 1997.

[Miyamoto, 1990] Sadaaki Miyamoto. *Fuzzy Sets in Information Retrieval and Cluster Analysis*. Kluwer Academic Publishers, Dordrecht, 1990.

[Mladenić and Grobelnik, 1998] Dunja Mladenić and Marko Grobelnik. Word Sequences as Features in Text-Learning. In *Proceedings of the 7th Electrotechnical and Computer Science Conference (ERK'98)*, pages 145–148, Ljubljana, Slovenia, 1998.

[Modha and Spangler, 2000] Dharmendra S. Modha and W. Scott Spangler. Clustering Hypertext with Applications to Web Searching. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, pages 143–152, San Antonio, TX, 2000.

[Modha and Spangler, 2003] Dharmendra S. Modha and W. Scott Spangler. Feature Weighting in k-Means Clustering. *Machine Learning*, **52**(3):217–237, 2003.

[Moore *et al.*, 1997] Jerome Moore, Eui-Hong (Sam) Han, Daniel Boley, Maria Gini, Robert Gross, Kyle Hastings, George Karypis, Vipin Kumar, and Bamshard Mobasher. Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering. In *7th Workshop on Information Technologies and Systems (WITS'97)*, Atlanta, GA, 1997.

[Muresan, 2002] Gheorghe Muresan. *Using Document Clustering and Language Modelling in Mediated Information Retrieval*. PhD thesis. School of Computing, The Robert Gordon University, Aberdeen, Scotland, 2002.

[Newton and O'Brien, 2002] Jack Newton and Christopher O'Brien. Scalable Clustering of Documents with Multiple Membership. Technical report. University of Alberta, Canada, 2002.

[Nielsen, 2003] Jakob Nielsen. Information Pollution. Jakob Nielsen's Alertbox, 11 August 2003.
URL (last visited on 17 January 2006): `www.useit.com/alertbox/20030811.html`.

[Noel *et al.*, 2003] Steven Noel, Vijay V. Raghavan, and Chee-Hung Henry Chu. Document Clustering, Visualization, and Retrieval Link Mining. In Wu *et al.* (2003), pages 161–194.

[Ogilvie and Callan, 2003] Paul Ogilvie and Jamie Callan. Combining Document Representations for Known-Item Search. In *Proceedings of the 26th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'03)*, pages 143–150, Toronto, Canada, 2003.

[Osdin *et al.*, 2002] Richard Osdin, Iadh Ounis, and Ryen W. White. Using Hierarchical Clustering and Summarisation Approaches for Web Retrieval: Glasgow at the TREC 2002 Interactive Track. In *Proceedings of the 11th Text REtrieval Conference (TREC 2002)*, Gaithersburg, MD, 2002.

[Page *et al.*, 1998] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-0120. Stanford Digital Library Technologies, 1998.

[Palmer *et al.*, 2001] Christopher R. Palmer, Jerome Pesenti, Raúl E. Valdés-Pérez, Michael G. Christel, Alexander G. Hauptmann, Dorbin Ng, and Howard D. Wactlar. Demonstration of Hierarchical Document Clustering of Digital Library Retrieval Results. In *Proceedings of the First ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL 2001)*, page 451, Roanoke, VA, 2001.

[Pantel and Lin, 2002] Patrick Pantel and Dekang Lin. Document Clustering with Committees. In *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 199–206, Tampere, Finland, 2002.

[Pekar *et al.*, 2004] Viktor Pekar, Michael Krkoska, and Steffen Staab. Feature Weighting for Co-occurrence-based Classification of Words. In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, pages 799–805, Geneva, Switzerland, 2004.

[Pelleg and Moore, 2000] Dan Pelleg and Andrew W. Moore. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pages 727–734, Stanford, CA, 2000.

[Pereira *et al.*, 1993] Fernando C. Pereira, Naftali Tishby, and Lillian Lee. Distributional Clustering of English Words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, 1993.

[Pericliev and Valdés-Pérez, 1998] Vladimir Pericliev and Raúl E. Valdés-Pérez. A Procedure for Multi-Class Discrimination and Some Linguistic Applications. In *Proceedings of the 17th International Conference on Computational Linguistics and of the 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL'98)*, pages 1034–1040, 1998.

[Perlich *et al.*, 2003] Claudia Perlich, Foster Provost, and Jeffrey S. Simonoff. Tree Induction vs. Logistic Regression: A Learning-curve Analysis. *Journal of Machine Learning Research*, **4**:211–255, 2003.

[Pirolli *et al.*, 1996] Peter Pirolli, James Pitkow, and Ramana Rao. Silk from a Sow's Ear: Extracting Usable Structures from the Web. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'96)*, volume 1, pages 118–125, Vancouver, Canada, 1996.

[Pitkow and Pirolli, 1997] James Pitkow and Peter Pirolli. Life, Death and Lawfulness on the Electronic Frontier. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'97)*, pages 383–390, Atlanta, GA, 1997.

[Porter, 1980] Martin F. Porter. An Algorithm for Suffix Stripping. *Program*, **14**(3):130–137, 1980. Reprinted in K. Sparck Jones and P. Willett, editors, *Readings in Information Retrieval*, pages 313–316, Morgan Kaufmann, San Francisco, 1997.

[Pullwitt and Der, 2001] Daniel Pullwitt and Ralf Der. Integrating Contextual Information into Text Document Clustering with Self-Organizing Maps. In *Advances in Self-Organising Maps, Proceedings of the Workshop on Self-Organising Maps (WSOM'01)*, pages 54–60, Lincoln, United Kingdom, 2001.

[Purandare and Pedersen, 2004] Amruta Purandare and Ted Pedersen. SenseClusters—Finding Clusters that Represent Word Senses. In *Proceedings of Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04)*, pages 26–29, Boston, MA, 2004.

[Renner and Aszódi, 2000] Alexander Renner and András Aszódi. High-throughput Functional Annotation of Novel Gene Products Using Document Clustering. In *Proceedings of the Fifth Pacific Symposium on Biocomputing*, pages 54–68, Honolulu, HI, 2000.

[Rezaee *et al.*, 1998] Mahmoud Ramze Rezaee, Boudewijn P.F. Lelieveldt, and Johan H.C. Reiber. A New Cluster Validity Index for the Fuzzy *c*-Mean. *Pattern Recognition Letters*, **19** (3–4):237–246, 1998.

[Riloff, 1995] Ellen Riloff. Little Words Can Make a Big Difference for Text Classification. In *Proceedings of the 18th International ACM Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 130–136, Seattle, WA, 1995.

[Roeck *et al.*, 2004a] Anne De Roeck, Avik Sarkar, and Paul Garthwaite. Defeating the Homogeneity Assumption. In *Proceedings of the 7th International Conference on the Statistical Analysis of Textual Data (JADT 2004)*, Louvain-la-Neuve, Belgium, 2004.

[Roeck *et al.*, 2004b] Anne De Roeck, Avik Sarkar, and Paul Garthwaite. Frequent Term Distribution Measures for Dataset Profiling. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, Lisbon, Portugal, 2004.

[Rose *et al.*, 2002] Tony G. Rose, Mark Stevenson, and Miles Whitehead. The Reuters Corpus Volume 1—from Yesterday's News to Tomorrow's Language Resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 29–31, Las Palmas de Gran Canaria, Spain, 2002.
URL (last visited on 17 January 2006): `http://about.reuters.com/researchandstandards/corpus/`.

[Rosell *et al.*, 2004] Magnus Rosell, Viggo Kann, and Jan-Eric Litton. Comparing Comparisons: Document Clustering Evaluation Using Two Manual Classifications. In *Proceedings of the International Conference on Natural Language Processing (ICON-2004)*, Hyderabad, India, 2004.

[Rosell, 2003] Magnus Rosell. Improving Clustering of Swedish Newspaper Articles using Stemming and Compound Splitting. In *Proceedings of the 14th Nordic Conference on Computational Linguistics (NoDaLiDa-2003)*, Reykjavik, Iceland, 2003.

[Rosenfeld and Morville, 1998] Louis Rosenfeld and Peter Morville. *Information Architecture for the World Wide Web*. O'Reilly and Associates, Sebastapol, CA, 1998.

[Rüger and Gauch, 2000] Stefan M. Rüger and Susan E. Gauch. Feature Reduction for Document Clustering and Classification. Technical Report DTR 2000/8. Department of Computing, Imperial College London, 2000.

[Sahlgren and Cöster, 2004] Magnus Sahlgren and Rickard Cöster. Using Bag-of-Concepts to Improve the Performance of Support Vector Machines in Text Categorization. In *Proceedings of the International Conference on Computational Linguistics (COLING 2004)*, pages 487–493, Geneva, Switzerland, 2004.

[Salton and Buckley, 1988] Gerard Salton and Christopher Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, **13**(5):513–523, 1988. Reprinted in K. Sparck Jones and P. Willett, editors, *Readings in Information Retrieval*, pages 323–328, Morgan Kaufmann, San Francisco, 1997.

[Salton and McGill, 1983] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[Saunders and Sheffield, 1998] Sam Saunders and Philip W. Sheffield. Searching for Clues—Education Professionals' Use of the Educati*on-line* Interface to Uncover Whole Texts in Education and Training. In *Proceedings of the Internet Research and Information for Social Scientists Conference (IRISS'98)*, Bristol, United Kingdom, 1998.

[Schmid, 1994] Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, Manchester, United Kingdom, 1994.

[Schmid, 1999] Helmut Schmid. Improvements in Part-of-Speech Tagging with an Application to German. In S. Amstrong, K.W. Church, P. Isabelle, S. Manzi, E. Tzoukerman, and D. Yarowsky, editors, *Natural Language Processing Using Very Large Corpora*, pages 13–25. Kluwer Academic Publishers, Dordrecht, 1999.

[Schütze and Silverstein, 1997] Hinrich Schütze and Craig Silverstein. Projections for Efficient Document Clustering. In *Proceedings of the 20th International ACM Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 74–81, Philadelphia, PA, 1997.

[Schütze, 1998] Hinrich Schütze. Automatic Word Sense Discrimination. *Computational Linguistics*, **24**(1):97–123, 1998.

[Shenk, 1997] David Shenk. *Data Smog: Surviving the Information Glut*. Harper Collins Publishers, New York, 1997.

[Shneiderman *et al.*, 1997] Ben Shneiderman, Donald Byrd, and W. Bruce Croft. Clarifying Search: A User-Interface Framework for Text Searches. *D-LIB Magazine of Digital Library Research*, 1997.
URL (last visited on 17 January 2006): `www.dlib.org`.

[Shneiderman, 1998] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, MA, 3rd edition, 1998.

[Silva *et al.*, 2001] Joaquim Silva, João Mexia, Carlos A. Coelho, and Gabriel Lopes. Multilingual Document Clustering, Topic Extraction and Data Transformations. In *Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving. Proceedings of the 10th Portuguese Conference on Artificial Intelligence (EPIA 2001)*, pages 74–87, Porto, Portugal, 2001.

[Singhal *et al.*, 1996a] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted Document Length Normalization. In *Proceedings of the 19th International ACM Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 21–29, Zurich, Switzerland, 1996.

[Singhal *et al.*, 1996b] Amit Singhal, Gerard Salton, Mandar Mitra, and Chris Buckley. Document Length Normalization. *Information Processing and Management*, **32**(5):619–633, 1996.

[Sinka and Corne, 2002] Mark Sinka and David Corne. A Large Benchmark Dataset for Web Document Clustering. In A. Abraham, J. Ruiz del Solar, and M. Koeppen, editors, *Soft Computing Systems: Design, Management and Applications*, pages 881–890. IOS Press, Amsterdam, 2002.

[Sinka and Corne, 2003a] Mark P. Sinka and David W. Corne. Towards Modernised and Web-Specific Stoplists for Web Document Analysis. In *Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI '03)*, pages 396–402, Halifax, Canada, 2003.

[Sinka and Corne, 2003b] Mark P. Sinka and David W. Corne. Evolving Better Stoplists for Document Clustering and Web Intelligence. In A. Abraham, M. Koeppen, and K. Franke, editors, *Design and Application of Hybrid Intelligent Systems*, pages 1015–1023. IOS Press, Amsterdam, 2003.

[Sinka and Corne, 2004] Mark P. Sinka and David W. Corne. The BankSearch Web Document Dataset: Investigating Unsupervised Clustering and Category Similarity. *Journal of Network and Computer Applications*, **28**(2):129–146, 2004.

[Skogmar and Olsson, 2002] Klas Skogmar and Johan Olsson. Clustering Documents with Vector Space Model using N-Grams. Course work, Lund Institute of Technology, Sweden, 2002.

[Slonim and Tishby, 2000] Noam Slonim and Naftanli Tishby. Document Clustering using Word Clusters via the Information Bottleneck Method. In *Proceedings of the 23rd International ACM Conference on Research and Development in Information Retrieval (SIGIR'00)*, pages 208–215, Athens, Greece, 2000.

[Slonim *et al.*, 2002] Noam Slonim, Nir Friedman, and Naftali Tishby. Unsupervised Document Classification Using Sequential Information Maximization. In *Proceedings of the 25th International ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 129–136, Tampere, Finland, 2002.

[Smeaton, 1997] Alan F. Smeaton. Information Retrieval: Still Butting Heads with Natural Language Processing? In M.T. Pazienza, editor, *Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology, International Summer School (SCIE-97)*, pages 115–138, Frascati, Italy, 1997.

[Smyth, 1996] Padhraic Smyth. Clustering Using Monte Carlo Cross-Validation. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 126–133, Portland, OR, 1996.

[Sneath and Sokal, 1973] Peter H.A. Sneath and Robert R. Sokal. *Numerical Taxonomy.* W.H. Freeman and Company, San Francisco, 1973.

[Spangler and Kreulen, 2002] W. Scott Spangler and Jeffrey Kreulen. Interactive Methods for Taxonomy Editing and Validation. In *Proceedings of the 11th International ACM Conference on Information and Knowledge Management (CIKM 2002)*, pages 665–668, McLean, VA, 2002.

[Sparck Jones, 1971] Karen Sparck Jones. *Automatic Keyword Classification for Information Retrieval.* Butterworths, London, 1971.

[Sparck Jones, 1999] Karen Sparck Jones. What is the Role of NLP in Text Retrieval? In Strzalkowski (1999), pages 1–24.

[Spink *et al.*, 2001] Amanda Spink, Dietmar Wolfram, Bernard J. Jansen, and Tefko Saracevic. Searching the Web: The Public and Their Queries. *Journal of the American Society for Information Science*, **52**(3):226–234, 2001.

[Stefanowski and Weiss, 2003] Jerzy Stefanowski and Dawid Weiss. Carrot^2 and Language Properties in Web Search Results Clustering. In *Web Intelligence: Proceedings of the First International Atlantic Web Intelligence Conference (AWIC 2003)*, pages 240–249, Madrid, Spain, 2003.

[Steinbach *et al.*, 2000] Michael Steinbach, George Karypis, and Vipin Kumar. A Comparison of Document Clustering Techniques. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD-00)*, Boston, MA, 2000. Extended version appears as Technical Report #00-034, University of Minnesota, 2000.

[Stevenson, 2003] Mark Stevenson. *Word Sense Disambiguation.* Center for the Study of Language and Information, Stanford, 2003.

[Strehl *et al.*, 2000] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of Similarity Measures on Web-page Clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI-2000)*, pages 58–64, Austin, TX, 2000.

[Strzalkowski, 1999] Tomek Strzalkowski, editor. *Natural Language Information Retrieval.* Kluwer Academic Publishers, Dordrecht, 1999.

[Sung *et al.*, 2003] Sam Yuan Sung, Zhao Li, and Tok Wang Ling. Clustering Techniques for Large Database Cleansing. In Wu *et al.* (2003), pages 227–260.

[Taylor, 1968] Robert S. Taylor. Question-Negotiation and Information Seeking in Libraries. *College & Research Libraries*, **29**(3):178–194, 1968.

[Tishby *et al.*, 1999] Naftali Tishby, Fernando C. Pereira, and William Bialek. The Information Bottleneck Method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, Urbana, IL, 1999.

[Titterington *et al.*, 1985] Donald Michael Titterington, Adrian F.M. Smith, and Udi E. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley and Sons, Chichester, 1985.

[Tombros and van Rijsbergen, 2001] Anastasios Tombros and Cornelis J. van Rijsbergen. Query-Sensitive Similarity Measures for the Calculation of Interdocument Relationships. In *Proceedings of the 10th International ACM Conference on Information and Knowledge Management (CIKM 2001)*, pages 17–24, Atlanta, GA, 2001.

[Tombros *et al.*, 2002] Anastasios Tombros, Robert Villa, and Cornelis J. van Rijsbergen. The Effectiveness of Query-Specific Hierarchic Clustering in Information Retrieval. *Information Processing and Management*, **38**(4):559–582, 2002.

[Tombros, 2002] Anastasios Tombros. *The Effectiveness of Query-Based Hierarchic Clustering of Documents for Information Retrieval.* PhD thesis. University of Glasgow, Scotland, 2002.

[Uchida *et al.*, 1999] Hiroshi Uchida, Meiying Zhu, and Tarcisio Della Senta. UNL: A Gift for a Millennium. Technical report. The United Nations University, Tokyo, Japan, 1999.

[Vakkari *et al.*, 1997] Pertti Vakkari, Reijo Savolainen, and Brenda Dervin, editors. *Information Seeking in Context. Proceedings of an International Conference on Research in Information Needs, Seeking and Use in Different Contexts, Tampere, Finland, 1996.* Taylor Graham, London, 1997.

[Vakkari, 1999] Pertti Vakkari. Task Complexity, Problem Structure and Information Actions—Integrating Studies on Information Seeking and Retrieval. *Information Processing and Management*, **35**:819–837, 1999.

[van Rijsbergen and Sparck Jones, 1973] Cornelis J. van Rijsbergen and Karen Sparck Jones. A Test for the Separation of Relevant and Non-Relevant Documents in Experimental Retrieval Collections. *Journal of Documentation*, **29**(3):251–257, 1973.

[van Rijsbergen, 1979] Cornelis J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

[Vinokourov and Girolami, 2000] Alexei Vinokourov and Mark Girolami. A Probabilistic Hierarchical Clustering Method for Organizing Collections of Text Documents. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'2000)*, volume 2, pages 182–185, Barcelona, Spain, 2000. Extended version available as Technical Report 5, University of Paisley, United Kingdom, 2000.

[Vivísimo, 2006] Vivísimo. Frequently Asked Questions. 2006.
URL (last visited on 17 January 2006): `http://vivisimo.com/html/faq`.

[Volk and Stepanov, 2001] Daniel Volk and Mikhail G. Stepanov. Resampling Methods for Document Clustering. Technical Report cond-mat/0109006. arXiv.org, 2001.
URL (last visited on 17 January 2006): `http://arxiv.org/pdf/cond-mat/0109006`.

[Volk, 1999] Martin Volk. Choosing the right lemma when analysing German nouns. In *Multilinguale Corpora: Codierung, Strukturierung, Analyse. 11. Jahrestagung der GLDV*, pages 304–310, Frankfurt, Germany, 1999.

[Voorhees, 1985] Ellen M. Voorhees. The Cluster Hypothesis Revisited. In *Proceedings of the 8th International ACM Conference on Research and Development in Information Retrieval (SIGIR'85)*, pages 188–196, Montreal, Canada, 1985.

[Voorhees, 1986] Ellen M. Voorhees. The Efficiency of Inverted Index and Cluster Search. In *Proceedings of the 9th International ACM Conference on Research and Development in Information Retrieval (SIGIR'86)*, pages 164–174, Pisa, Italy, 1986.

[Wang and Kitsuregawa, 2001] Yitong Wang and Masaru Kitsuregawa. Link Based Clustering of Web Search Results. In *Proceedings of the Second International Conference on Advances in Web-Age Information Management (WAIM 2001)*, pages 225–236, Xi'an, China, 2001.

[Wang and Kitsuregawa, 2002] Yitong Wang and Masaru Kitsuregawa. On Combining Link and Contents Information for Web Page Clustering. In *Proceedings of the 13th International Database and Expert Systems Applications Conference (DEXA-2002)*, pages 902–913, Aix-en-Provence, France, 2002.

[Ward, 1963] Joe H. Ward. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, **58**(301):236–244, 1963.

[Weil and Rosen, 1997] Michelle M. Weil and Larry D. Rosen. *TechnoStress: Coping With Technology @Work @Home @Play*. John Wiley and Sons, New York, 1997.

[Weiss *et al.*, 1996] Ron Weiss, Bienvenido Vélez, Mark A. Sheldon, Chanathip Namprempre, Peter Szilagyi, Andrzej Duda, and David K. Gifford. HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering. In *Proceedings of the 7th ACM Conference on Hypertext*, pages 180–193, Washington, DC, 1996.

[Weiss *et al.*, 2000a] Sholom M. Weiss, Brian F. White, and Chidanand Apte. Lightweight Document Clustering. Technical Report RC-21684. IBM T.J. Watson Research Center, 2000.

[Weiss *et al.*, 2000b] Sholom M. Weiss, Brian F. White, Chidanand V. Apte, and Frederick J. Damerau. Lightweight Document Matching for Help-Desk Applications. *IEEE Intelligent Systems*, **15**(2):57–61, 2000.

[Welch, 1984] Terry A. Welch. A Technique for High-Performance Data Compression. *IEEE Computer*, **17**(6):8–19, 1984.

[Wen and Zhang, 2003] Ji-Rong Wen and Hong-Jiang Zhang. Query Clustering in the Web Context. In Wu *et al.* (2003), pages 195–226.

[Wen *et al.*, 2001] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering User Queries of a Search Engine. In *Proceedings of the 10th International World Wide Web Conference (WWW-10)*, pages 162–168, Hong Kong, China, 2001.

[Wilbur and Sirotkin, 1992] W. John Wilbur and Karl Sirotkin. The Automatic Identification of Stop Words. *Journal of Information Science*, **18**(1):45–55, 1992.

[Willett, 1985] Peter Willett. Query-specific Automatic Document Classification. *International Forum on Information and Documentation*, **10**(2):28–32, 1985.

[Willett, 1988] Peter Willett. Recent Trends in Hierarchical Document Clustering: A Critical Review. *Information Processing and Management*, **24**(5):577–597, 1988.

[Williams, 2000a] Simon Williams. A Survey of Natural Language Processing Techniques for Text Data Mining. Technical Report 2000/127. CSIRO Mathematical and Information Sciences, Australia, 2000.

[Williams, 2000b] Simon Williams. A Survey of Text Data Mining Products. Technical Report 2000/128. CSIRO Mathematical and Information Sciences, Australia, 2000.

[Winkle, 1998] William Van Winkle. Information Overload: Fighting data asphyxiation is difficult but possible. *Computer Bits magazine*, 1998. 1 February 1998.

[Wu *et al.*, 2003] Weili Wu, Hui Xiong, and Shashi Shekhar, editors. *Clustering and Information Retrieval*. Kluwer Academic Publishers, Dordrecht, 2003.

[Xiao, 2003] Yangzhe Xiao. Evaluation of Kernel Function Modification in Text Classification Using SVM. Language Workshop of the First instructional/informational Conference on Machine Learning (iCML-2003), Rutgers University, Piscataway, NJ, 3–8 December 2003.

[Xu and Croft, 1996] Jinxi Xu and W. Bruce Croft. Query Expansion Using Local and Global Document Analysis. In *Proceedings of the 19th International ACM Conference on Research and Development in Information Retrieval (SIGIR'96)*, pages 4–11, Zurich, Switzerland, 1996.

[Yang and Pedersen, 1997] Yiming Yang and Jan O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML'97)*, pages 412–420, Nashville, TN, 1997.

[Yang, 1997] Shu Ching Yang. Information Seeking as Problem-Solving Using a Qualitative Approach to Uncover the Novice Learners' Information-Seeking Processes in a Perseus Hypertext System. *Library & Information Science Research*, **19**(1):71–92, 1997.

[Yang, 1999] Yiming Yang. An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval*, **1**(1-2):69–90, 1999.

[Zadeh, 1965] Lotfi A. Zadeh. Fuzzy Sets. *Information and Control*, **8**(3):338–353, 1965.

[Zamir and Etzioni, 1998] Oren Zamir and Oren Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st International ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 46–54, Melbourne, Australia, 1998.

[Zamir and Etzioni, 1999] Oren Zamir and Oren Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. *Computer Networks*, **31**(11–16):1361–1374, 1999. See also *Proceedings of the 8th International World Wide Web Conference (WWW-8)*, Toronto, Canada, 1999.

[Zamir, 1999] Oren Zamir. *Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results*. PhD thesis. University of Washington, 1999.

[Zhang *et al.*, 1996] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. In *Proceedings of the 1996 ACM International Conference on Management of Data (SIGMOD-96)*, pages 103–114, Montreal, Canada, 1996.

[Zhao and Karypis, 2001] Ying Zhao and George Karypis. Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report TR #01-040. University of Minnesota, 2001.

[Zhao and Karypis, 2002] Ying Zhao and George Karypis. Evaluation of Hierarchical Clustering Algorithms for Document Datasets. In *Proceedings of the 11th International ACM Conference on Information and Knowledge Management (CIKM 2002)*, pages 515–524, McLean, VA, 2002. Appears also as Technical Report TR #02-22, University of Minnesota, 2002.

[Zhao and Karypis, 2003] Ying Zhao and George Karypis. Hierarchical Clustering Algorithms for Document Datasets. Technical Report TR #03-027. University of Minnesota, 2003.

[Zhong and Ghosh, 2005] Shi Zhong and Joydeep Ghosh. Generative Model-based Clustering of Documents: a Comparative Study. *Knowledge and Information Systems (KAIS)*, **8**:374–384, 2005.

[Zhou and Zhang, 2003] Lina Zhou and Dongsong Zhang. NLPIR: A Theoretical Framework for Applying Natural Language Processing to Information Retrieval. *Journal of the American Society for Information Science and Technology*, **54**(2):115–123, 2003.

[Ziv and Lempel, 1977] Jacob Ziv and Abraham Lempel. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory*, **23**(3):337–343, 1977.

Acknowledgements

I would like to express my deep gratitude to all those who have actually read this thesis in full and have offered constant support and invaluable advice: Prof. Michael Hess and Prof. Avi Bernstein. Special thanks go to the *Neue Zürcher Zeitung* (NZZ) and the *Schweizerische Depeschenagentur* (SDA), which have kindly provided two excellent corpora for use in this study. For technical support, guidance and patience I would like to thank in particular Beat Rageth, Manfred Klenner and Simon Clematide of the Department of Informatics and the Institute of Computational Linguistics at Zurich University. Other colleagues and office friends who have supported me technically or morally during these three years are too numerous to list here individually—to all of them a heartily felt "thank you". Finally, I cannot end without gratefully acknowledging the unfailing support and generous encouragement that I have received from my family, from my friends and in particular from Silvia.

Curriculum Vitae

Richard Forster

Born 18 January 1975 in Winterthur. Attended primary school and the *gymnasium* "Rychenberg" in Winterthur (1982–1994). From 1994 to 2001 studied Wirtschaftsinformatik (information science and economics) at Zurich University, graduating with a diploma thesis in Artificial Intelligence under Prof. Rolf Pfeifer ("Visualising Artificial Ontogeny—Grovis"). Study was interrupted by military service and combined with various activities as an international chess master, trainer and writer. Since 1996 chess editor of the *Neue Zürcher Zeitung* and contributor to nearly a dozen international magazines. From 1998 to 2003 first occasional then full-time research for *Amos Burn—A Chess Biography*, published in 2004 by McFarland Inc., Jefferson NC, USA (972 pp.). From 2003 to 2006 work on the present PhD thesis under Prof. Michael Hess at the Institute of Computational Linguistics in Zurich. Since 2005 work on a new chess history project for the 200th anniversary of the Schachgesellschaft Zürich in 2009.